Next-Best-View Planning for Exploration and Autonomous 3D-Modeling in Static Environments with Irregular Depth Noise using Interval Probabilities

Alexandra von Lösecke

DLR

<table>
<tr><td rowspan="2">Institut für Robotik und Mechatronik</td><td>BJ.: 2015</td></tr>
<tr><td>IB.Nr.: 572-2015/29</td></tr>
</table>

MASTERARBEIT

NEXT-BEST-VIEW PLANNING FOR EXPLORATION AND AUTONOMOUS OBJECT MODELLING IN STATIC ENVIRONMENTS WITH IRREGULAR DEPTH NOISE USING INTERVAL PROBABILITIES

Freigabe:	Der Bearbeiter:	Unterschriften
	Alexandra von Lösecke	A. v. Lösecke
	Betreuer:	
	Christian Rink	
	Der Institutsdirektor	
	Dr. Alin Albu-Schäffer	Albu-Schäffer

Dieser Bericht enthält 66 Seiten, 30 Abbildungen und 5 Tabellen

Ort: Oberpfaffenhofen	Datum:	Bearbeiter:	Zeichen:

Technische Universität München
Institute for Media Technology
Prof. Dr.-Ing. Eckehard Steinbach

Deutsches Zentrum für Luft- und
Raumfahrt e. V.
Institute of Robotics and Mechatronics

Master Thesis

Next-Best-View Planning for Exploration and Autonomous 3D-Modelling in Static Environments with Irregular Depth Noise using Interval Probabilities

Author:	Alexandra von Lösecke
Matriculation Number:	03617617
Address:	Neustätter Straße 16a
	82327 Tutzing
Supervising Professor:	Prof. Dr.-Ing. Eckehard Steinbach
Advisor:	Christian Rink
Begin:	18.05.2015
End:	18.11.2015

Kurzfassung

Diese Arbeit untersucht next-best-view Planung für Exploration und autonome 3D-Modellierung in statischen, verrauschten Umgebungen unter Verwendung von Intervallwahrscheinlichkeiten.
Ein Algorithmus zur autonomen Exploration wird erweitert, um mit verrauschten Tiefendaten aus Innenraum-Umgebungen besser zurechtzukommen. Die aufgenommenen Daten werden von einem neuartigen Voxel Space-Update interpretiert, welches ein aktuell verwendetes Bayes-Update auf dreidimensionalen Besetztheitskarten mit dem Konzept der Intervallwahrscheinlichkeiten kombiniert. Damit wird eine zusätzliche Informationsebene der Unsicherheit eingeführt. Der Raum wird mit Hilfe eines angepassten Next-Best-View Kriteriums exploriert, welches die maximale Entropie jedes Volumenelements berechnet.
In einer Simulationsumgebung werden Experimente durchgeführt, um zu untersuchen wie aus den Möglichkeiten der Wahrscheinlichkeitsintervalle ein Nutzen gezogen werden kann. Die Ergebnisse werden mit einer Implementierung des Bayes-Update am DLR-RMC verglichen.
Die Methodik wird dann auf die Objektmodellierung übertragen. Mit einem industriellen KR16 Roboterarm des Herstellers KUKA werden weitere Experimente durchgeführt, um zu prüfen, ob die Ergebnisse aus der Simulation auf die Realität übertragbar sind. Die bedeutendste Schlussfolgerung aus den Experimenten ist, dass das neuartige Update besonders im Umgang mit widersprüchlichen Daten vorteilhaft ist.

Abstract

This thesis examines Next-Best-View planning for exploration and autonomous 3D-modelling in static environments with depth noise using Interval Probabilities.
An autonomous exploration algorithm is extended to cope better with depth noise from non-diffusely reflecting environments. The measured data is interpreted by a novel voxel space update which combines a state-of-the-art Bayes Update on three dimensional occupancy maps with the concept of Interval Probabilities. Thereby an additional informational level of uncertainty is introduced. An adapted next-best-view criterion, which calculates the maximum entropy of each volume element, aids the exploration process.
Experiments are conducted in a simulation to examine how to take advantage of the possibilities of Interval Probabilities. The results are compared to an implementation of the Bayes Update at the DLR-RMC. The method is then transferred to object modelling. A robotic industrial arm named KUKA KR16 is used to conduct further experiments to verify whether the results from the simulation are applicable to real-life environments. It is concluded that the novel update is especially advantageous in dealing with specular reflections.

Contents

Chapter 1

Introduction

1.1 Motivation

Industry as well as society has come to rely more and more on robots. Going by the name “robota”, Czech for “forced labour” [Dic], their purpose is to serve. Ever since the term was introduced in the K. Čapek’s play “Rossum’s Universal Robots”, robots have been constructed for a wide array of activities, which all have the one objective of providing service to humans. Often, they perform tasks we as humans can not or do not want to do ourselves: They assemble cars, they lift and weld mechanical parts for us or they do our housework when we let them vacuum our apartment. In storage buildings, they organize and redistribute goods.

When robots begin work on tasks that were meant to be done by humans, they have to deal with a human environment. The hallways might be narrow, long, and featureless, there are stairs. Glass and mirrors are used as design elements. This environment has not been designed for robots, as they are, for instance, hardly able to correctly interpret mirrors as surfaces. As we integrate robots into our everyday life, we will encounter such challenges over and over. Thus, we need to teach robots how to cope with such circumstances.

A good starting point for that task is to collect information about the situation and interpret it, e.g. to create a map or a model. However, the information can be misinterpreted or it can be insufficient. If decisions are based on such a questionable level of information, the result can be anything from illogical to dangerous. Therefore putting the collected information into context is crucial. If utilized well, it can avoid misinterpretation and provide additional information. One way of establishing a context around the collected information is through introducing an additional level of uncertainty, taking into consideration how much information has already been collected about the event. In this thesis, this additional uncertainty is introduced through the usage of interval probabilities. When assuming a probability for an information to be true, the certainty about that probability

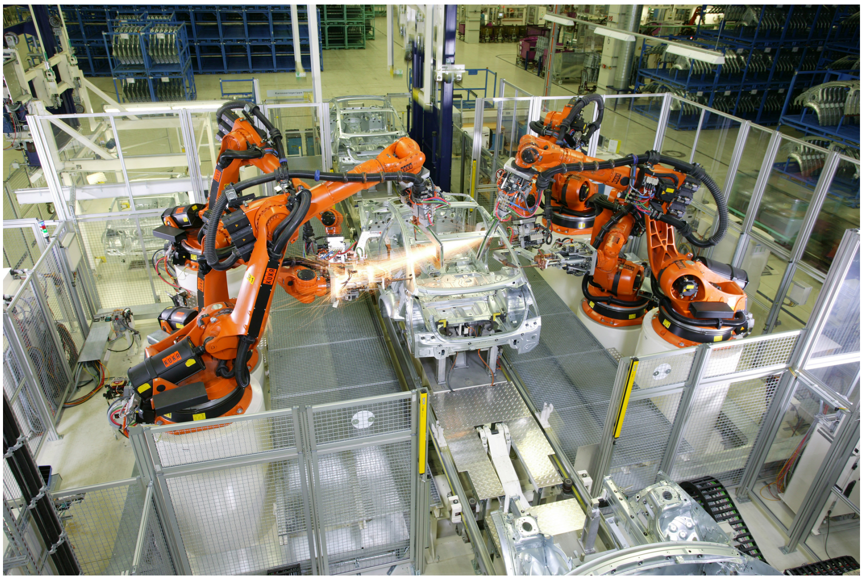


Figure 1.1: Robotic arms weld parts of a car. Photo courtesy of KUKA Robotics [KUKa]

is represented by an interval.
In this thesis, the concept of interval probabilities is applied to autonomous exploration and object modelling. The goal is to improve the interpretation of perceived data which is erroneous or conflicting and thus enable the robot to decide on better actions.

1.2 Problem Description: Autonomous Exploration in Environments with Depth-Noise

As the name suggests, autonomous exploration requires autonomy, i.e. independence and the capability and cognition to explore. The problem of exploration has been thoroughly discussed in the literature, for instance by Thrun, Burgard and Fox [TBF05]. They explain that when exploring, a robot should "maximize its knowledge about the external world". This is achieved through implementing intelligent algorithms on robots, enabling them to perceive, interpret, decide and act autonomously. The state-of-the-art exploration algorithms usually follow four steps:

- Perception
- Interpretation
- Decision
- Action

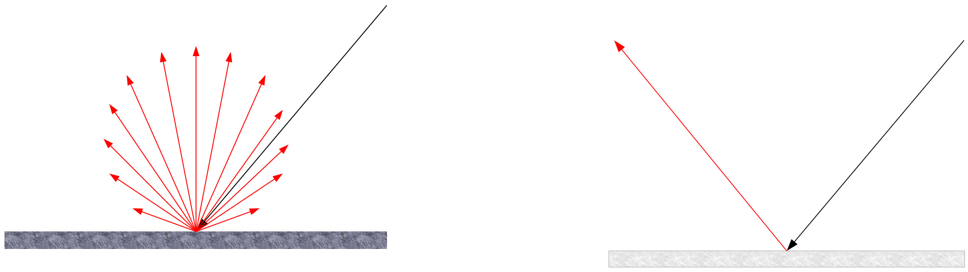
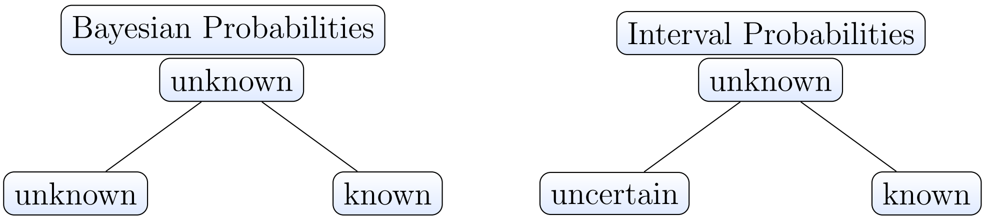


Figure 1.2: Diffuse (left) and specular (right) reflection. The incoming beam is black, while the reflected beams are coloured red.

Firstly, the exploration algorithm of an intentional system should gather data using its sensors. From the collected data, it is able to infer conclusions, e.g. create a map. It can then decide on actions based on those inferences, depending on how to best pursue a given objective. Lastly, the actions that have been chosen are carried out. I.e., the robot should analyse measured data, decide on a next-best-view and move to it such that a given goal is achieved while taking certain demands into consideration. A robot finding its way through a maze, for instance, would first measure the walls that surround him, analyse where it can go and then decide where it should go depending on what looks most promising.
For the exploration purposes, movement is restricted to two dimensions, since the laboratory does not include stairs nor any other kind of change in altitude. Later, when the developed approach is transferred to object modelling, movement in the third dimension offers significant information gain and can thus not be omitted.

Conflicting data can impede the exploration process. In state-of-the-art procedures, some kind of 3D imaging technique is commonly used. The imaging process relies on the event that a beam emerging from the camera is directly reflected towards the sensor. This only happens on diffusely reflecting surfaces. As described by Juds [Jud88], diffuse surfaces reflect light towards all directions. Thus, some light is directly reflected towards the sensor. On other surfaces, the beam is not reflected and the distance measurement is distorted. Specularly reflecting surfaces, for instance, reflect most of the light in one direction. Unless the surface is viewed from a 90° angle, the angle of incidence will not equal the angle of reflection. Thus, the light has to be be reflected by at least one other object in order to be perceived by the sensor. The measured distance will thus not correspond to the actual distance between the object and the sensor. Other phenomena like absorption and transparency further decrease the intensity of light reflected directly towards the sensor. In this thesis, the phenomenon that light is not reflected diffusely due to transparency, absorption or specular reflectance is referred to as depth noise. In human indoor environments, reflecting, absorbing and transparent surfaces are a frequent occurrence. If perception is erroneous, an interpretation based solely on that measurement



(a) After information has been collected, states are unknown or known

(b) After information has been collected, states are uncertain or known

Figure 1.3: Bayesian Probabilities vs. so-called Imprecise Probabilities: States resulting from conflicting data can be distinguished with Imprecise Probabilities, while the state remains unknown in certain cases of Bayesian Probabilistics.

is error prone as well. Thus, in the state-of-the-art exploration algorithm, information from several scans is collected and combined in a representation of the space that is being explored. In a perfect environment, the space that is initially unknown eventually becomes known. However, since an indoor environment designed for humans is not perfect, certainty can not be attained about every area. When using the Bayes update, these areas remain unknown, making them non distinguishable from unperceived voxels. This likely hinders coping with conflicting data, as a conflict can not be identified and is thus not targeted specifically.

Bayesian Probabilistics are limited in the way that they require a probability for each event in question. This means that neither initial ignorance nor uncertainty about information can be modelled. Many authors such as Dempster and Shafer [Sha92] and Walley [Wal91] have tried to bypass these limitations by introducing uncertainty of some kind. Based on the work of these authors, this thesis will introduce Interval Probabilities as a measure of uncertainty to allow a distinction between ignorance, i.e. absence of information, and uncertainty about existing information, i.e. failure to unequivocally interpret existing data. The essential difference between the so-called Interval Probabilities that allow for a distinction and Bayesian Probabilities is clarified in Fig. 1.3. Since uncertainty in spite of existing data is likely due to conflicting data, identifying uncertainties can help identify conflicts. Prospective algorithms will be able to decide on actions to properly cope with the conflict and ideally resolve it.

In summary, this thesis aims at enabling exploration in non-ideal environments by planning next-best-views under consideration of uncertainties in the space representation. In the future, it is hoped that the idea can be expanded to cope with conflicting data independent of the specific situation.

Multiple reasons can motivate the usage of imprecise probabilities. There are practical reasons like robustness of conclusions drawn from a statistical analysis, but there is also the desire to represent real world situations. For instance, precise probabilities are not sufficient when modelling the beliefs of a group of people. The motives for usage of imprecise probabilities are described by Walley [Wal91]. Below is a summary of the motives most relevant to this thesis:

A new, unknown object about to be scanned can be easily represented through vacuous probabilities. Rather than assuming a precise probability that is likely inaccurate due to absence of data, using imprecise probabilities means we can assume complete initial ignorance. This can be interpreted as a boundary condition representing the model's starting point.

Since the objective of this thesis is to model unknown areas or objects, the fact that shapes are unknown but become known needs to be represented appropriately. Situations in which the occupancy of a cell is uncertain in spite of data being available need to be distinguished from cells which have not been scanned. This is achieved through imprecise probabilities. The amount of information available is thus modelled more accurately. In contrast to the previously mentioned assumption of initial ignorance, the amount of information as obtained throughout several exploration steps is modelled.

Since the given task of deciding whether or not a cell is occupied is quite a complex one, determining the probability that a cell is occupied will be complex as well. Finding a single precise value of this probability may not be realistic, since this requires knowledge of the situation as well as time and computational power. Instead, upper and lower bounds are introduced and assessed.

When modelling the beliefs of a group of people or any other set of differing beliefs, imprecise probabilites can merge the divergent opinions into one model. For instance, if person A believes the probability of event X is 0.8, and person B believes that event will occur with a probability of 0.5, an interval ranging from 0.5 to 0.8 can represent both beliefs at once. The same applies to robotic exploration. In the given task, the sensor will rotate around the object, scanning it from different angles and positions. Thus, many cells will be scanned more than once, each scan resulting in a different probability of occupancy. This data can be merged into a single hypothesis through the usage of imprecise probabilities. Convergent data can be interpreted as highly consistent, whereas non-convergent data can mean either inconsistency or lack of knowledge.

1.3 Structure

Chap. 2 gives an overview of the state-of-the-art of autonomous exploration and introduces Interval Probabilities. In Chap. 3, the exploration strategy that this thesis is based on is introduced. The exploration process is extended in Chap. 4 by introducing a novel update type, the so-called Interval Update. A possibility to exploit the advantages of the Interval Update is suggested and implemented. Experiments are conducted to examine how the usage of interval probabilities affects performance, both in autonomous exploration and object modelling. The results are described in Chap. 5. Chap. 6 sums up the thesis and its results and gives an outlook on future work on the topic.

Chapter 2

Related Work

2.1 Data structure

When perceiving the environment with depth sensors, the data needs to be represented in an internal structure. The most straightforward approach when measuring with time-of-flight cameras is to convert the data to a point cloud. However, other structures such as elevation maps [HCK+89], multi-level surface maps [TPB06] or occupancy grids [WHB+10][Sup08] are also commonly used.
An occupancy grid represents the environment like a map. It divides the given space along a grid and describes the occupancy of the resulting subspaces. This is especially helpful when exploring environments, since the occupancy grid can be used for mapping as well as for planning the next step. As described in the PhD thesis by Suppa [Sup08], a so-called three-dimensional (3D) voxel space is often generated during the scanning process. The space that is to be scanned is divided into cubic subspaces, also named volumetric elements or so-called voxels. As with any occupancy grid, each subspace is assigned a probability of occupancy and a corresponding state, e.g. “free”, “occupied” or “unknown”. In an octree, groups of eight voxels are named children and assigned a superordinate parent node, while eight of those parents are again assigned a superordinate node. Child nodes are pruned if they are stable and have the same probability of occupancy.

2.2 Exploration Strategies in Comparison

2.2.1 State-of-the-Art of Autonomous Exploration

In order to be able to assess and evaluate the exploration strategy in use at the German Aerospace Center, Deutsches Zentrum für Luft- und Raumfahrt (DLR)-Robotics and

Mechatronics Center (RMC)-Department of Perception and Cognition (PEK), other published exploration strategies are examined. Chen et al. give an overview of various perception strategies in their paper: Active vision in robotic systems: A survey of recent developments [CLK11]. They describe two major issues of Next-Best-View (NBV) planning:

- Navigating safely in spite of the fact that the environment is partially unknown and the sensors' capabilities are limited
- Ensuring a sufficient overlap of the current map and the local model.

Farshidi et al. [FSK09]use two statistical metrics for rating a scan position: mutual information and the Cramer-Rao lower bound.

In another paper, Holz and Amigoni present "A comparative evaluation of exploration strategies and heuristics to improve them" [HBAB11]. The authors compare three exploration strategies:
The "Closest Frontier Exploration Strategy" [Yam97] detects borders between explored and unknown regions and selects the frontier closest to the current position.
Gonzalez Banos and Latombe‘s Exploration Strategy [GBL02] evaluates utility $u(p)$ by taking travelling cost $L(p)$ and information gain $I(p)$ into account and thereby selecting the Next-Best-Scan (NBS) position:

$$u(p) = I(p) \cdot e^{-\lambda L(p,r)}. \tag{2.1}$$

The information gain is given by the "difference [in] entropy before(H) and after($\hat{H}$) the scan"(p.3)

$$I(p) = \hat{H} - H \tag{2.2}$$

$$H = -\sum_{c^{[xy]}} \left[p(c^{[xy]}) log p(c^{[xy]}) + (1 - p(c^{[xy]})) log(1 - p(c^{[xy]})) \right] \tag{2.3}$$

The authors thereby tackle the problem of balancing the requirements named in Chen et al. article.
Another approach, the Multi-Criteria Decision Making (MCDM) - based Exploration Strategy [BA09] uses the following criteria for decision making:

- travelling cost $L()$
- estimated information gain $I()$
- overlap $O()$

The utility of each criterion $i, i \in \{L(), I(), O()\}$ is then calculated as follows:

$$u_L(p) = \frac{1 - (L(p,r) - \min_{q \in C} L(q,r))}{\max_{q \in C} L(q,r) - \min_{q \in C} L(q,r)} \tag{2.4}$$

and the utilities are combined by a Choquet fuzzy integral:

$$u(p) = \sum_{i=1}^{|N|} (u_{(i)}(p) - u_{(i-1)}(p))\mu(A_i) \tag{2.5}$$

Furthermore, the authors make suggestions for improving the strategies above. They claim that an improvement with respect to travelling cost can be achieved by introducing Repetitive Re-checking and Map Segmentation. Repetitive re-checking means that the map is updated during navigation and the selected frontier is constantly challenged. If the frontier becomes known during navigation to the chosen location, navigation is terminated and a new frontier is selected. When using map segmentation, the map is divided into segments. Frontiers inside the segment of the robots location are preferred to those in other segments.

2.2.2 Research on Autonomous Exploration and 3D Modelling at DLR-RMC

The following section describes research conducted at DLR-RMC [DLR] in the field of autonomous exploration and 3D modelling. Kriegel et al. present their various techniques they use when scanning objects. They focus on model completeness as well as scene exploration with multiple objects.
One of their first approaches [KRB+12] is to determine NBS candidates from boundary search. The NBS position is selected using the Information Gain IG, which is based on the sum of the entropies of all voxels visible from the sensor in the NBS position:

$$IG_{scan} = \sum_{beams} \sum_{voxels} H_{voxel}(p) \tag{2.6}$$

$$H_{voxel}(p) = -\underbrace{p\log(p)}_{occupied} - \underbrace{(1-p)\log(1-p)}_{free}. \tag{2.7}$$

with H_{voxel} describing the entropy of a voxel and p being the probability of occupancy of that voxel. Later, after a certain estimated coverage of the mesh is reached, a hole detection algorithm is applied. I.e. NBS positions are planned based on the location and size of holes in the mesh.
In a recent paper([KRBS13]), that approach is modified: Again, NBS positions are selected by analysing a volumetric model of the unknown object. In this case, however, the selection of a scan position is based on a utility function:

$$f_{utility} = \underbrace{(1-\omega)\cdot e_v}_{Exploration} + \underbrace{\omega\cdot(1-q_s)}_{3D\ Modeling} \tag{2.8}$$

using the entropy of volumetric model e_v, the weighting ω, and a surface quality value q_s. q_s is calculated from the local sampling density, the incidence angle, and the amount of border edges. The scan position with the highest utility value is selected for the next scan. With the same approach as in the paper published in 2012 [KRB+12], NBS candidates are found and sorted, except that scenes are examined instead of individual objects ([KBM+13]). Therefore, a sensor pose that is not in collision with other objects and with a minimally occluded view needs to be chosen. Similarly to the authors' article in 2013 [KRBS13], a utility function f is used. In this case, however, an actual function is given to evaluate the surface quality:

$$q_s = \frac{1}{k}\sum_{i=1}^{k} \lambda \cdot b_i + (1-\lambda) \cdot d_i, \qquad \lambda, b_i, d_i \in [0,1]. \tag{2.9}$$

Also, the weight is defined:

$$\omega = \frac{\frac{n_s}{n_q}}{\left(\frac{n_s}{n_q}+1\right)} \tag{2.10}$$

with n_s being the scan number, whereas n_q is chosen and in this case set to 8.

An earlier approach by the authors [KBSH11] is similar to their other works, considering boundary search is again used. The new aspect in this case is the classification of found edges into left, right, top and bottom edges and the fact that rightmost, leftmost, lowermost and topmost boundaries are prioritized, respectively, when dealing with multiple edges in a single scan.

The strategy by Kriegel et al.[KRBS13] has also been utilized by Thomas et al. [TKS14]. They select scan path candidates by extending the algorithm, i.e. using boundary detection with surface trend estimation and, later, hole detection. Just like in Kriegel et al.'s approach [KRBS13], a mesh and a voxel space are constructed. Once both the mesh and the voxel space are completed, scans are performed on a cylinder around the object in order to obtain color information. Additionally, the object is rotated and its bottom part is scanned.
An autonomous exploration system at the DLR-RMC has been introduced by Suppa and Hirzinger [SH07]. A multipurpose vision system is obtained by mounting multiple sensors to a robot. The robot is thus enabled to explore an entirely or partially unknown environment by building a map incrementally.

2.3 Coping with Conflicting Data in Exploration and 3D-Modelling

Alt et al. [ARS13] present a method to reconstruct transparent objects with a Kinect sensor. First, the sensor moves around the scene containing the transparent object and creates a background model. Then a model consisting of model depth $\overline{D_c}$, reliability R_c, and the standard deviation value$\sigma'_c = \max(\sigma_c, \sigma_s)$, obtained by taking the maximum of the observed value σ_c and the expected value σ_s for the standard deviation. From these values, a so-called error signal is calculated. This value allows detection of transparent as well as reflecting areas. Experiments show that a location as well as a rough shape estimate can be obtained. According to the authors, the approach is limited to surfaces with dominantly refractive effects, i.e. objects with smooth and curved surfaces.

Another approach to coping with noisy data is presented by Magnusson et al. [MLD07]. They examine scan registration for autonomous mining vehicles, by applying a normal distributions transform. In a NDT, the surface is represented by normal distributions, combined to give insight on the probability that a surface point is present at the corresponding location. Thus, a continuous, piecewise differentiable representation is modelled, making the nearest neighbour search dispensable. This is helpful when dealing with rough surfaces, which produce seemingly conflicting, but matchable data. The major challenge in this case is matching point clouds, while the noisiness of the data is rather secondary. Its applicability to situations where the central task is coping with noise has yet to be proved.

The presence of specular reflections is often noted ([HHHL+14]) as a source of noise and errors, but a strategy to cope with the noise is scarcely suggested. Hansard et al. [MHH12] examine the performance of Time-of-Flight(ToF) cameras when measuring specularly reflecting and translucent surfaces. Since the book is meant to give an overview over ToF-cameras, it does not suggest a practical solution to this issue. An early approach suggested by Yamauchi [Yam97] is the combination of sonar and laser sensors, the so-called laser-limited sonar. This only considers the noise from the sonar sensor but neglects the noise from the laser sensor.

2.4 Interval Probabilities

Bayesian Probabilistics are limited in the way that they require a probability for each event in question. The probability of an event A and the probability of a complementary event $\overline{A}$ always add up to one. A number of other authors have managed to bypass this limitation by introducing uncertainty. Dempster [Dem68] and Shafer [Sha76], for instance, use uncertainty to provide a measure of probability of one event based on the probability for another event. They call this measure a degree of belief. The degree of belief for an event could be based on the probability of reliability of a person telling you about that event. Moreover, Dempster has stated a rule for combining beliefs if they are based on independent evidence. Thus, in contrast to Bayesian Probabilistics, when using Dempster-Shafer belief functions, the degree of belief for an event A and the degree of belief for a complementary event $\overline{A}$ do not necessarily add up to one. Authors like Suppa [Sup08] have implemented the approach into a so-called Belief Update, which uses Dempster's rule of combination instead of the commonly used Bayes Update to update cells.

Another approach by Walley [Wal91] is motivated by gambles: The author defines so-called lower previsions as the maximum price at which buying a gamble is desirable. An upper prevision, in contrast, is the minimum price at which selling a gamble is desirable. In Bayesian Probabilistics, the lower prevision and the corresponding upper prevision match, i.e. the buying price equals the selling price. The inconsistency in this model can be illustrated by interpreting gambles as goods bought and sold by a merchant. The merchant is usually not willing to buy goods at the same price at which he wants to sell them. Thus the lower and upper previsions differ. Many situations are better modelled through imprecision.

Lower previsions by themselves do not guarantee an advantage, though. Three principles need to be followed: Avoiding sure loss, coherence, and natural extension. Avoiding sure loss means not accepting a series of gambles which degrades your situation. Moreover, a gamble whose outcome is better or equally as good as set of gambles that are already accepted, should be accepted as well, i.e. beliefs should be coherent. Lastly, if a set of gambles exist, coherent inferences are desirable. This can be achieved by natural extension of the existing beliefs. If beliefs are extended under consideration of avoiding sure loss, this automatically leads to coherent probabilities. Thus, natural extension can be used to combine different assessments into a coherent model. Interval Probabilities take these three principles into consideration and are therefore well-suited to combine different measurements into a coherent 3D-model.

The idea of imprecision has been implemented into classification trees. Abellan and Moral [AM01] introduce a total uncertainty criterion. By considering a combination of the maximum entropy and a measure for the total uncertainty, they reduce entropy while not increasing nonspecificity. In a later publication, they state that a credal set's "uncertainty can be measured by considering maximum entropy" [AM03].

Chapter 3

Fundamentals of Autonomous Exploration at DLR

An overview of the most important methods for the implementation of this thesis are described in the following. As the name suggests, autonomous exploration requires autonomy, i.e. independence and the capability and cognition to explore. This is achieved through implementing intelligent algorithms on robots, enabling them to perceive, interpret, decide and act autonomously.

3.1 Autonomous Robots

When it comes to three-dimensional scanning and exploration, the goal is to eventually create a robotic intentional system in order to replace manual scanning by a "human intentional system". Firstly, the exploration algorithm of an intentional system should be able to infer conclusions from previously gathered information. It can then decide on actions based on those inferences, all the while pursuing a given objective. Lastly, the actions that have been chosen are carried out. I.e., the robot should analyse measured data and decide on a next-best-view such that a given goal is achieved while taking certain demands into consideration. E.g. a robot finding its way through a maze would first measure the walls that surround him, analyse where it can go and then decide where it should go depending on what looks most promising.

Figure 3.1: Room in which exploration will be performed

3.2 Perception

A typical laboratory environment as existent at DLR is displayed in Fig. 3.1. The walls border an environment consisting of larger interior objects such as desks and closets, as well as smaller items such as a trash can. Some of the passages are broad enough so the robot can drive past them, others are narrowed by obstacles. In this setup, every element diffusely reflects the sensor beams, such that the measurement data is noise-free.

In order to perceive such an environment, robots need sensors. Autonomous robots are often equipped with odometers and optical sensors. While the odometry data is mainly used for position estimation, optical data can additionally be utilized for various purposes, including mapping, obstacle avoidance and object recognition. Nevertheless, odometry data is crucial, since an approximate knowledge of the robot's position is needed to match measured datasets to known data.

In this thesis, the perception is seen as a circumstance that can not be influenced. The emphasis is on the interpretation of the perceived data rather than on finding the optimal sensor for the task. Choosing a different, optimal sensor for each task and mounting all of the chosen sensors to the robot would not be resource conserving. Moreover, the task of determining the type and number of appropriate sensors is complex in itself and thus not considered in this thesis.

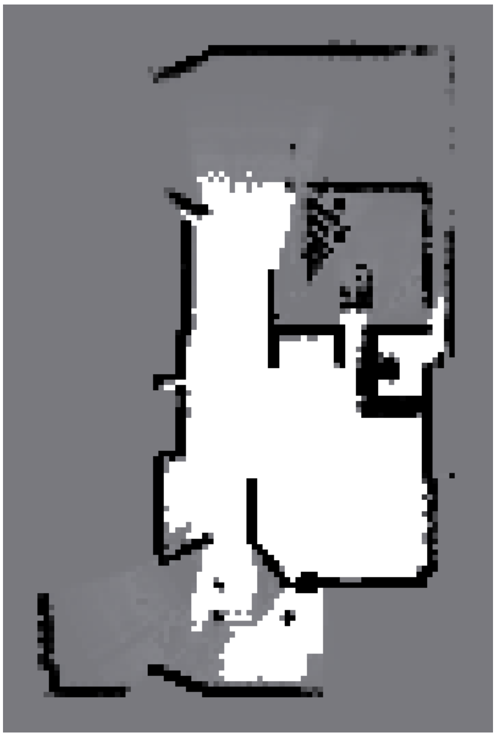


Figure 3.2: A typical voxel map. Voxels with a high probability of occupancy are coloured black, while voxels whose state is unknown are gray. Voxels that are likely free are transparent, thus the white background is visible.

3.3 Interpretation

Once measurement data is perceived, it is interpreted by the robot as explained in the following. First, the robot locates itself within the map using odometry data as well as the obtained depth images. For details on this procedure, the interested reader is referred to Wohlfahrt's Thesis [Woh15]. Once the position is known, the depth images are mapped to an occupancy grid. Details about this space representation are given in the following.

3.3.1 Creation of an Occupancy Map

The structure of the state-of-the-art octree has already been explained in Sect. 2.1. An exemplary visualization of an octree can be seen in Fig. 3.2. Occupied voxels are black, while free voxels are transparent. Gray voxels are unknown. The picture shows how the sensors have already detected obstacles such as walls and office furniture, while the space that has not been perceived by the sensors is still unknown. The image also shows why the sensors do not perceive their entire environment at at once: In some cases, the environment is occluded by obstacles, in other cases the environment is out of reach to the sensors, since

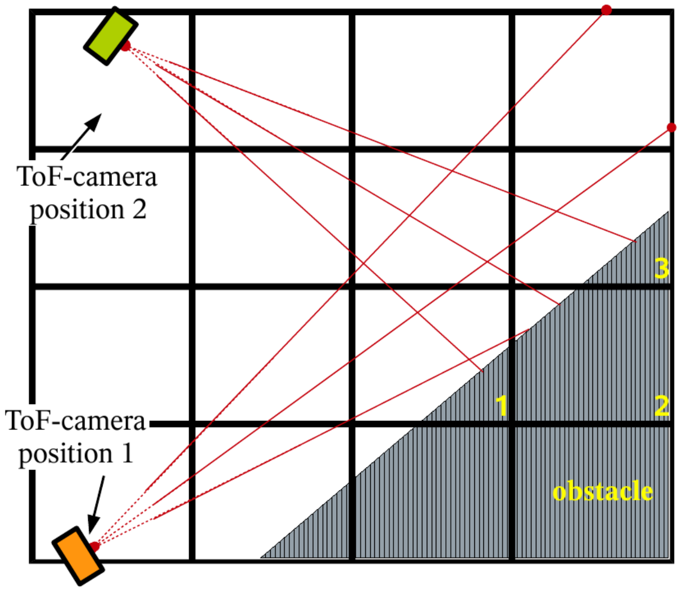


Figure 3.3: Depending on the scanning angle, an obstacle is perceived differently

they can only measure reliably up to a distance of a few metres. Moreover, the number and arrangement of sensors does usually not allow for a complete 360° view. Nevertheless, the robot is able to fully explore its environment by performing additional measurement steps from different viewpoints.
The voxel space's resolution can be defined as the number of points within a three-dimensional space that can be distinguished. In a voxel space, it is equivalent to the number of voxels per spatial unit. Thus, the smaller the edge length of a voxel, the larger the resolution. Choosing an appropriate resolution for the purpose is crucial: If using a resolution that is too coarse, the accuracy of the map decreases. If it is chosen too fine, the performance of the algorithm suffers, since more voxels need to be iterated for the same size of space. As displayed in Fig. 3.3, there are two features of a voxel space that should be considered.

- The voxel space does not represent the environment perfectly, unless the resolution is infinitely high, or the objects' outlines perfectly match the grid. A higher resolution usually leads to a better representation of the environment. For exploration purposes, a coarse grid is especially precarious. Fig. 3.3 shows that once an obstacle is perceived, some free space becomes inaccessible, since the entire voxel is considered to be occupied. The amount of inaccessible free space is linked to the grid size. This means a higher resolution is advantageous.

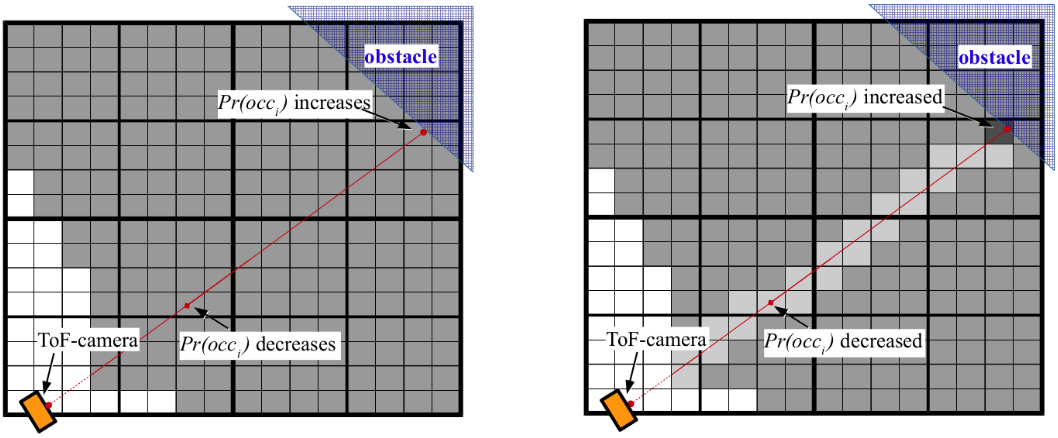
- Moreover, when interpreting a measurement, the scanning angle should be taken into consideration. If a surface is viewed from a flat angle, the measurement beams might pass by the surface. As illustrated in Fig. 3.3, the flat scanning angle of the

ToF-camera in position 1 is inferior to the near perpendicular angle from position 2. The beams emerging from the camera in position 1 pass through voxels **1** and **3** even though they contain obstacles, thereby giving the impression that those voxels are free. If the robot relies on this information, it may collide with the obstacle. In contrast, the beams emerging from the camera in position 2 hit the obstacle in all three voxels, giving the impression that those three voxels are occupied. This information will keep the robot from colliding with the obstacle.

The initial uncertainty about the map or object is typically modelled by setting the probability of occupancy to 0.5 in each voxel. By analysing the amount and location of unknown voxels, the position and angle of the next scan, the so-called "next-best-view", can be planned (see subsection 3.3.3). As more and more scans are performed, the state of each voxel ideally changes from "unknown" to either "free" or "occupied", such that finally the state of all voxels is known. With sufficient resolution, the occupied voxels imitate the shape of the scanned object, or trace the walls of the explored room. In some cases, the voxel space is used exclusively for next-best-view planning. In other cases, a mesh is additionally constructed from the scan data. This is especially important if the goal is to construct a 3D-model, and can be omitted in exploration applications for the sake of performance.
Since the position of the robot is known, the depth images from the eight ToF-cameras can be positioned within the voxel space. The part of the depth images that makes up the floor is cut off and not considered when building the voxel space, since the floor is occupied, but does not pose an obstacle. The voxels' states can then be updated using the information from the depth images. The update strategy for updating each voxel state is explained in the following.

3.3.2 Bayes Update

The Bayes update is a kind of iteration that allows to infer conclusions about the state of voxels from measurement data.
Each beam is traced from the sensor's light source to the supposed obstacle. All of the voxels that are intersected by the trace are considered for the Bayes update. This is illustrated on the left hand side of Fig. 3.4. The farthest voxel which is intersected by the beam contains the polygon which poses the obstacle. Knowing that the sensor has just found an obstacle in that voxel, its probability of occupancy is increased. Similarly, if a voxel's probability of occupancy suggests that it is occupied, but the measurement beam passes through it, the probability of occupancy can be decreased to display the information gain from the latest measurement.
This means that if the probability of occupancy after several measurements is high, an obstacle has repeatedly been detected in that voxel, while a low probability of occupancy means the beam passed through that voxel several times. If the probability of occupancy



(a) Voxel space before the update (b) Voxel space after the update

Figure 3.4: Applying the Bayes update to a voxel space

remains at 0.5, the area has either not been perceived by the sensor or the measurement data is inconsistent, i.e. the voxel appears to be both occupied and free, depending on the scan.

Decreasing and increasing the probability value is performed by a Bayes Updater [Sti86][Joy08]:

The updater is based on Bayes' theorem, which states that the probability of an event A to occur given the event B, $Pr(A|B)$, can be calculated from the probability of an event B given A, $Pr(B|A)$, if the probabilities of the individual events, $Pr(A)$ and $Pr(B)$, are known.

$$Pr(A|B) = \frac{Pr(B|A) \cdot Pr(A)}{Pr(B)} \tag{3.1}$$

This can be rephrased and applied to the problem of finding the probability of occupancy of a voxel:

$$Pr(Occ_i|Data) = \frac{Pr(Data|Occ_{i-1}) \cdot Pr(Occ_{i-1})}{Pr(Data)} \tag{3.2}$$

In this interpretation, Bayes' theorem acts as an iteration step, where the above formula describes step i. It transforms the probability that a voxel is occupied, $Pr(Occ_{i-1})$, to the probability of occupancy given a certain additional knowledge, or data. $Pr(Occ_i|Data)$ is calculated from the likelihood of the data, taking into consideration the knowledge about the occupancy of the voxel. The theorem also considers the overall probability of the data, which acts as a scaling factor. The next iteration step $i+1$ is performed once new data is available.

$Pr(Occ_{i-1})$ is referred to as A-priori-probability, whereas $Pr(Occ_i|Data)$ is called the A-posteriori-probability. Even though $Pr(Data)$ is merely a scaling factor, it is difficult to calculate, since every possible measurement has to be taken into consideration. In order to avoid having to quantify $Pr(Data)$, one can take advantage of the fact that Equation 3.2

can be applied to the probability of vacancy $Pr(Free)$ as well.

$$Pr(Free_i|Data) = \frac{Pr(Data|Free_{i-1}) \cdot Pr(Free_{i-1})}{Pr(Data)} \tag{3.3}$$

Solving Equation 3.3 for $Pr(Data)$

$$Pr(Data) = \frac{Pr(Data|Free_{i-1}) \cdot Pr(Free_{i-1})}{Pr(Free_i|Data)} \tag{3.4}$$

and inserting it into Equation 3.2 eliminates $Pr(Data)$. This is called an odds-representation of Bayes' theorem. Odds are a way of expressing how strongly two events are associated with each other.

$$Pr(Occ_i|Data) = \frac{Pr(Data|Occ_{i-1})}{Pr(Data|Free_{i-1})} \cdot \frac{Pr(Occ_{i-1})}{Pr(Free_{i-1})} \cdot Pr(Free_i|Data) \tag{3.5}$$

$$\underbrace{\frac{Pr(Occ_i|Data)}{Pr(Free_i|Data)}}_{:=\text{ o(i)}} = \underbrace{\frac{Pr(Data|Occ_{i-1})}{Pr(Data|Free_{i-1})}}_{:=\text{ lq}} \cdot \underbrace{\frac{Pr(Occ_{i-1})}{Pr(Free_{i-1})}}_{:=\text{ o(i-1)}} \tag{3.6}$$

The result is a posterior odd $o(i)$, which is created from an updated prior odd $o(i-1)$ using the likelihood quotient lq.
In the next calculation step $i+1$, the former posterior odd becomes the prior odd. Applying the logarithm to both sides of the equation leads to a simple summation:

$$log(o(i)) = log(lq \cdot o(i-1)) = log(lq) + log(o(i-1)), \tag{3.7}$$

assuming $Pr(Occ_i|Data)$, $Pr(Data|Occ_{i-1})$ and $Pr(Occ_{i-1})$ are greater than zero. This is commonly called the log-odds representation. To retrieve the posterior probability of occupancy, the fact that $Pr(Free)$ can be calculated from $Pr(Occ)$ is exploited:

$$Pr(Free) = 1 - Pr(Occ) \tag{3.8}$$

$$\log(o(i)) = \log(\frac{Pr(Occ_i)}{Pr(Free_i)}) = \log(\frac{Pr(Occ_i)}{1 - Pr(Occ_i)}), \tag{3.9}$$

$$o(i) = \frac{Pr(Occ_i)}{Pr(Free_i)} = \frac{Pr(Occ_i)}{1 - Pr(Occ_i)}, \tag{3.10}$$

the posterior probability of occupancy can be retrieved from the posterior odd by reversing Equation 3.10:

$$Pr(Occ_i) = \frac{o(i)}{1 + o(i)}. \tag{3.11}$$

Combining all of the calculations above results in the following mapping relation for updating the prior probability $Pr(Occ_{i-1})$:

$$Pr(Occ_i) = \frac{\frac{Pr(Occ_{i-1})}{1-Pr(Occ_{i-1})} \cdot lq}{1 + \frac{Pr(Occ_{i-1})}{1-Pr(Occ_{i-1})} \cdot lq}. \tag{3.12}$$

$Pr(Occ)$	< 0.05	> 0.95	$0.05 < Pr(Occ) < 0.95$
set to state	FREE	OCCUPIED	unchanged(UNKNOWN)
set to $Pr(Occ)$	0	1	unchanged

Table 3.1: State adjustments depending on $Pr(Occ)$

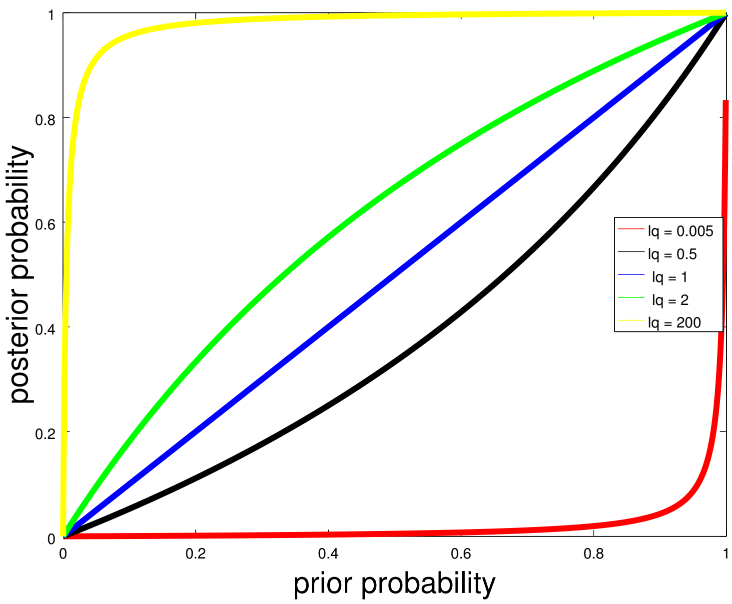


Figure 3.5: Mapping from prior to posterior probability depending on the likelihood quotient

If the probability of occupancy of a voxel passes the threshold for being occupied or being free, the state is adjusted accordingly. As listed in Tab. 3.1, if the probability of occupancy of an individual cell drops below 0.05, the cell's state is set to FREE. Similarly, the state is set to OCCUPIED if the $Pr(Occ)$ increases past 0.95.

The effect of the likelihood quotient is visualized in Fig. 3.5, which shows the update from the prior probability $Pr(Occ_{i-1})$ to the posterior probability $Pr(Occ_i)$.

It can be seen that a likelihood quotient smaller than 1 maps the prior probability to a lower posterior probability, i.e. it decreases the probability of occupancy. Similarly, a likelihood quotient greater than 1 maps the prior probability to a higher posterior probability, increasing the probability of occupancy. This effect is intuitive if the likelihood quotient is examined closely: The numerator contains the probability of measuring this data in the examined voxel, assuming it is occupied. The denominator contains the opposite, i.e the probability of measuring this data, assuming the voxel is free. If it is more likely that the data occurred due to an occupied than to a free voxel, the likelihood quotient is larger than 1. The voxel is probably occupied and the probability of occupancy should be increased.

This analogously applies if the likelihood quotient is smaller than 1.
Another feature of the update is that once a voxel's state reaches the numerical limits of its type, the update has no effect on it any more. The numerical limits of the chars stored in the voxel space are 0 and 255, which, converted to double for the update step, results in limits of 0.0 and 1.0.

$$Pr(Occ_i)|_{Pr(Occ_{i-1})=0.} = \frac{\frac{0}{1-0} \cdot lq}{1 + \frac{0}{1-0} \cdot lq} = 0 \tag{3.13}$$

$$Pr(Occ_i)|_{Pr(Occ_{i-1})=1.} = \lim_{Pr(Occ_{i-1}) \to 1.} \frac{\frac{Pr(Occ_{i-1})}{1-Pr(Occ_{i-1})} \cdot lq}{1 + \frac{Pr(Occ_{i-1})}{1-Pr(Occ_{i-1})} \cdot lq} = 1. \tag{3.14}$$

From the updated voxel space, an exploration map can be created as explained in the following section.

3.3.3 Designing an Exploration Map

As explained earlier, the movement of autonomous robotic platforms considered in this thesis is limited to two dimensions. Thus, while the data contained in third dimension is crucial to exploration itself, i.e. to its purpose of discovering and collecting information, it is hardly relevant for planning exploration steps. Thus the three-dimensional voxel space is transformed into a two-dimensional (2D) so-called exploration map. This decreases computational effort significantly when pursuing objectives such as finding frontiers and path planning. The 2D map is transformed into an exploration map designed for pursuing such objectives as explained in the following. A detailed explanation and motivation for the usage of an exploration map is described by Wohlfahrt [Woh15].
As a preparatory step for creating the exploration map, the 3D voxel space is projected along its z-axis onto a 2D map. Since the voxels that represent the floor have already been excluded, they are likewise not considered in the projected map. For each column of voxels along the z-axis, a pixel is created on the map. If the column contains one or more occupied voxels, the pixel is considered occupied, as the area represented by the pixel contains an obstacle. If all of the voxels' states in one column are free, the pixel is considered free as well and the area represented by the pixel is possibly reachable.
Once the occupied areas are determined, i.e. obstacles are identified, the robot needs to plan its next step. Before planning where it should go, delimiting the areas it can go is important.
Since the robot should be able to rotate freely at any time, its geometry needs to be taken into account. The omniRob has a cuboid shape and therefore requires a sufficient cylindrical obstacle free area in order to rotate without collision. The radius of this cylinder defines the safety distance that the robot needs to keep to all obstacles. This area is called collision space in the following. As a result, passages may appear too narrow to navigate for the robot even though it could pass them at the right rotation. However, navigation is

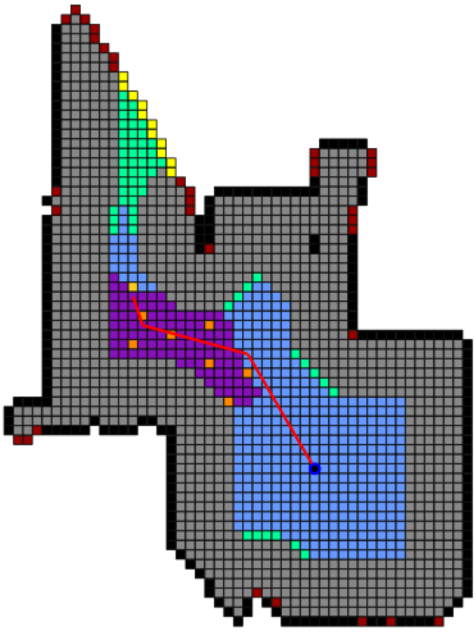


Figure 3.6: Exploration map created from voxel space

colour		meaning
blank		unknown area
black		occupied
gray		collision space
blue		free, reachable
dark red		unknown voxel
light green		maybe collision space
dark green		unreachable maybe collision space
light red		border
light yellow		selected frontier
purple		scan area around frontier
orange		sampled robot positions
yellow		selected robot position

Table 3.2: Colors in the exploration map and their meanings

not the main area of concern in this thesis. Therefore creating a collision space is a handy simplification.

Once a 2D-representation of the voxel space has been created, the exploration map as displayed in Fig. 3.6 is designed from the information contained in that representation.
As summarized in Fig. 3.2, the exploration map is a color coded 2D-representation of the voxel map. The area that is still unknown is blank. This usually means the voxels are either too far away from the sensor or hidden due to an obstacle. If information is available about an area, it is coloured as follows: Occupied pixels pose obstacles and are painted black. As explained previously, the robot's rotation is not considered and it needs to keep a certain distance to all obstacles. This collision space is marked in gray in the exploration map. Space which is not collision space and which is reachable from the robot's current position is considered reachable free space, marked in blue.
Voxels that lie within the collision space, but whose state is unknown, are coloured dark red. Such voxels may occur if the omniRob's sensors have not perceived that particular voxel yet, or if the measurements are inconsistent. A so-called maybe-collision space (light green) needs to be created around them for safety, even though navigating through the area is uncritical if the unknown voxel is free. Since these voxels can pose an issue to the exploration algorithm, their number should be reduced as much as possible. The fewer unreachable voxels are unknown, the smaller the chance a maybe collision space obstructs the path.
In order to plan the next exploration step, the target of that next step is determined.

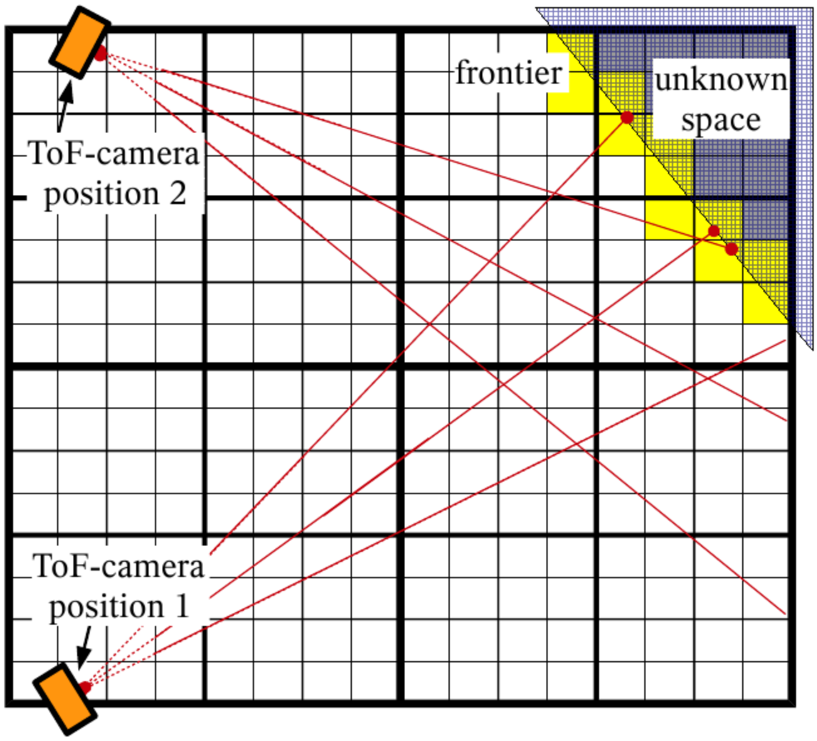


Figure 3.7: Determining the information gain of each point of view

Since the information gain from scanning a large amount of unknown voxels is likely to be high, any reachable area bordering an unknown area is considered worth exploring. Out of all the borders(light red) that fulfil this condition, the closest one is chosen as the next frontier(yellow).
A circular area in appropriate distance to frontier is marked in purple color in the exploration map. NBV-positions are sampled within that area. The decision process between the sampled candidates is described in the following section.

3.4 Decision: Next-Best-View Selection at DLR

Several state-of-the-art decision strategies have been examined in Sect. 2.2. The strategy used for exploration at DLR-RMC is explained in the following: As depicted in Fig. 3.7, different next-best-view candidates, or viewpoints, are considered. For each viewpoint, the paths of all sensor beams are simulated. Each sensor beam passes through a number of voxels before it either hits an obstacle or reaches the maximum measuring distance. In the figure, two of the beams emerging from sensor 1 pass through a free area and enter an unknown area. Occupied and free voxels are not taken into account when calculating the entropy, since their entropy is close to zero. Thus, the entropy is only calculated for the voxels within the unknown area. The same applies to the ToF-Camera in position 2. Assuming that the probability of occupancy in each frontier and unknown voxel is

$$Pr(Occ) = p = 0.5, \tag{3.15}$$

as it is usually the case when using the Bayes update, the entropy H as defined by Shannon [SW48] of an individual voxel is calculated as:

$$H = -(p \cdot \log(p) + (1-p) \cdot \log(1-p)) \tag{3.16}$$

From that, the entropy of a set of voxels can be calculated by summing up the entropies of all voxels in the set:

$$H = \sum_{i=1}^{\substack{\text{num of}\\\text{voxels}}} -(Pr(Occ_i) \cdot \log(Pr(Occ_i)) + (1 - Pr(Occ_i)) \cdot \log(1 - Pr(Occ_i))) \tag{3.17}$$

The robot position from which the voxels with the highest sum of entropies can be viewed is selected as the next-best-view. The robot then plans its path to that position and moves accordingly. A new exploration step as described in the previous sections is started.

3.5 Termination

Once all reachable unknown voxels have been explored, no new frontier and no new next-best-view can be found. Therefore, the exploration process is terminated. Ideally, the environment is now fully explored and the exploration map is a representation of the entire environment.

Chapter 4

Interval Update

The previously described exploration algorithm uses Bayesian probabilistics. Another approach is the usage of Interval Probabilities. Similar approaches are the theory of belief and imprecise probabilities. The usage itself as well as a reasoning is explained in the following.

4.1 Basic Concept

Bayesian probabilistics are commonly used in a wide array of robotic applications. This has hardly been questioned by researchers, since the uncertainty modelled by Bayesian probabilistics is often sufficient. However, when confronted with non-ideal environments which inadvertently occur in real-life situations, an additional level of uncertainty may be desirable.
According to the ideas of Dempster and Shafer as introduced in Chap. 2, knowing the probability for an event A to occur does not induce any knowledge about the complementary event $\overline{A}$. Instead, uncertainty is modelled through interval probabilities.
In Bayesian probabilistics, each voxel is assigned a state, which is an interpretation of its probability of occupancy. When using interval probabilities, each voxel is assigned not only a state, but also an interval of upper and lower bounds of that state. The lower bound $\underline{Pr}(Occ)$ is the lowest probability of occupancy that can be justified with data. Analogously, the upper bound $\overline{Pr}(Occ)$ is the highest probability of occupancy which can be assumed for that voxel. As described in Tab. 4.1, the same can be said about the probability of vacancy $Pr(Free)$: The lower bound $\underline{Pr}(Occ)$ is the lowest justifiable probability that a voxel is free. Since the voxel can only be free or occupied, the probability that either of the two occurs is 1. In this model, the state is calculated from the interval bounds and thereby becomes a mere interpretation of the interval. Using the average value as the state may seem obvious, but using the upper or lower interval bound or any other value calculated from the interval as the state is just as consequential. Using

	upper bound	lower bound
occupied	$\overline{Pr}(Occ)$	$\underline{Pr}(Occ) = 1 - \overline{Pr}(Free)$
free	$\overline{Pr}(Free)$	$\underline{Pr}(Free) = 1 - \overline{Pr}(Occ)$
uncertain(both free and occupied)	1	1

Table 4.1: Expressions for upper and lower bounds for different states of a voxel

the lower interval bound as the state is an optimistic approach: It is initially assumed that all surrounding areas are free and the robot can go anywhere. Obstacles are only considered if they have been perceived by the robot. Moreover, obstacles are only seen as obstacles if the lower interval bound, i.e. the state, exceeds the occupancy threshold: $Pr(Occ) > 0.95$.
In contrast, using the upper interval bound as the state is a pessimistic approach which assumes a worst-case scenario. Initially, the entire environment is assumed occupied: $Pr(Occ) = 0.95$. The robot can not go anywhere, except if measurements lower the upper interval bound below the threshold for free voxels: $Pr(Free) = 0.05$.

4.2 Derivation of Interval Update from Bayes Update

The State-of-the-Art Bayes update as described in subsection 3.3.2 is already in use in several applications at DLR. In order to cope with the noise from measuring non-ideal surfaces, the update is to be applied to a probability interval. This replaces the usual single value for the probability of occupancy P_{occ} of a voxel. The interval consists of an upper and lower bound for the probability of occupancy.

The probability interval in use consists of a lower bound for the probability of occupancy $\underline{Pr}(Occ)$ and a lower bound for the probability of vacancy $\underline{Pr}(Free)$. Equation 3.8 needs to be rephrased accordingly:

$$\overline{Pr}(Free) = 1 - \underline{Pr}(Occ) \tag{4.1}$$

$$\overline{Pr}(Occ) = 1 - \underline{Pr}(Free) \tag{4.2}$$

The posterior odds for the interval bounds are calculated in analogy to Equation 3.6.

$$o(i) = \frac{\underline{Pr}(Occ_i)}{\overline{Pr}(Free_i)} = \underbrace{\frac{Pr(Data|Occ_{i-1})}{Pr(Data|Free_{i-1})}}_{likelihood\ quotient} \cdot \underbrace{\frac{\underline{Pr}(Occ_{i-1})}{\overline{Pr}(Free_{i-1})}}_{prior\ odd\ o(i-1)} \tag{4.3}$$

Over time, the two probabilities $\underline{Pr}(Occ_i)$ and $\underline{Pr}(Free_i)$ will approach each other. If the data is consistent, the interval will converge towards an occupied or free state. Otherwise,

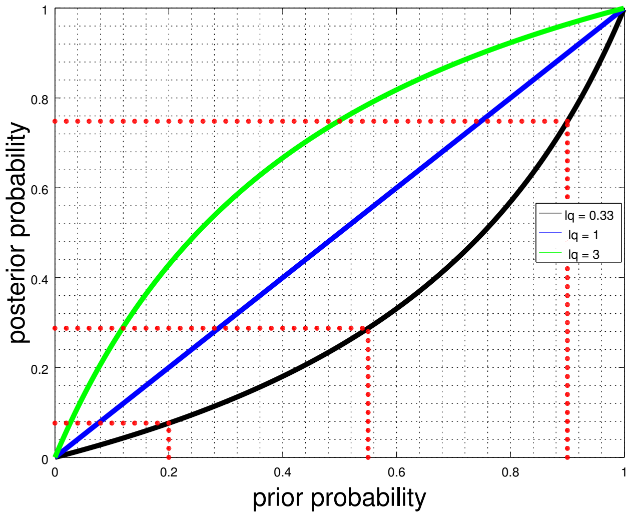


Figure 4.1: Updating a probability interval vs. updating a probability value

the interval will narrow towards varying values. If the data is consistent and the interval is narrow enough to fall below a lower threshold or exceed an upper threshold, the cell's state is adjusted.

The previously introduced interval update does need a little more storage space, since two double values need to be stored to define each voxel's probability of occupancy instead of one. However, chances are that the additional information will bring advantages in exploration performance as well as a better insight about the current situation of the robot.
Moreover, an update step will have less effect on the state than with the Bayes update. As explained in subsection 3.3.2, the update function is not linear, unless the likelihood quotient equals 1. Furthermore, the relation between the state and the interval bounds is often defined linearly, e.g. if the state is set to the mean of the interval limits. This results in a rather inert update compared to the Bayes update.
An exemplary update step is illustrated in Fig. 4.1. Assuming the interval bounds before the update are $I_{i-1} = [0.2, 0.9]$, the state, defined as the mean, equals 0.55. If a state of $s_{i-1} = 0.55$ is updated by a Bayes update with a likelihood quotient of $lq = 0.33$, the state is mapped to approximately $s_i = 0.3$. In contrast, updating each interval bound separately and then calculating the mean results in a state of approximately $s_i = 0.4$, which is much closer to the original value of 0.55. An Interval Update will therefore require more steps for the state to reach the threshold for defining the voxel as occupied or free. Whether or not this difference is beneficial for the exploration process will be discussed in Chap. 5.

4.3 Application of Imprecise Probabilities

In the following section, the properties of the previously introduced interval update are examined. A strategy on how to take advantage of the additional information is suggested.

4.3.1 Advantages of the Interval Update

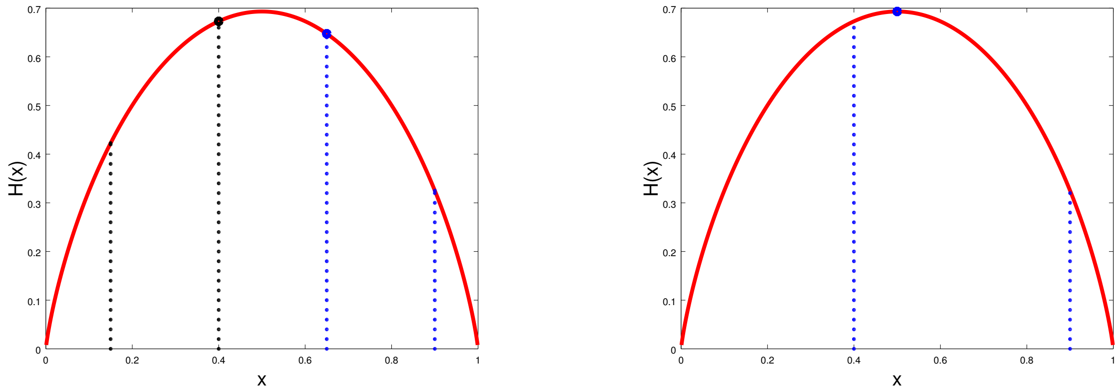
The update type introduced in Chap. 4 does need more storage space, since two double values need to be stored to define each voxel's probability of occupancy instead of one. However, the lack of information is modelled more precisely using probability intervals. Describing the knowledge about the cell through intervals means that an additional level of uncertainty is modelled.
Assuming a certain value for the probability of occupancy is unsubstantiated unless it is supported by data. This ignorance, or lack of knowledge, is modelled through a large probability interval. Once data is collected, the certainty about the assumed probability value increases. This is modelled through a narrowing interval. Chances are that the additional information and computational effort will bring advantages in exploration performance as well as a better insight about the current situation of the robot. A suggestion on how to exploit the properties of the newly introduced update type is detailed in the following: Using a maximum entropy criterion for decision on NBV. The method is implemented and empirically investigated in Chap. 5.

4.3.2 Decision on Next-Best-View by Maximum Entropy

As explained in Sect. 3.4, once a frontier has been selected, a good scan position is sought. The reachable area in appropriate viewing distance to the frontier and possible scan positions are determined. When using the Interval Update, the information gain from each scan position is determined as explained in Sect. 3.4. This is sufficient when using a Bayes Update. However, using the interval update, another strategy better exploits the advantages of that update type. (Fig. 4.2a - Fig. 4.2b). The figures display different positions of the interval bounds relative to the entropy. As suggested in [AM03], minimizing the maximum of entropy is a promising strategy. In each exploration step, the maximum of entropy in each cell is determined. Depending on the location of the interval bounds, a different value is used for calculating the entropy.
According to Shannon [SW48], the entropy H is calculated as:

$$H = \sum_{i=1}^{\substack{\text{num of}\\ \text{voxels}}} -(Pr(Occ_i) \cdot \log(Pr(Occ_i)) + (1 - Pr(Occ_i)) \cdot \log(1 - Pr(Occ_i))) \tag{4.4}$$

Since the entropy function is symmetrical and strictly monotonic within $[0; 0.5[$ and $]0.5; 1]$, it reaches its maximum at 0.5. The entropy function and some exemplary intervals are



(a) Maximum at upper or lower interval bound, depending on which is closer to $Pr(Occ) = 0.5$

(b) Maximum of entropy at 0.5

Figure 4.2: The calculation of the maximum of entropy depends on the position of interval bounds. The black and the blue interval in Fig. 4.2a do not cover the value 0.5. The function is thus evaluated at the bound closest to 0.5. In contrast, the maximum entropy of the interval in Fig. 4.2b is reached at 0.5.

displayed in Fig. 4.2a and Fig. 4.2b. The value of maimum entropy is marked with a bold dot. The interval coloured in black in Fig. 4.2a lies entirely below the point at which the entropy function reaches its maximum. Conversely, the blue interval lies entirely above that point. Thus, if

$$0.5 \notin [a, b], a, b \in [0, 1], a < b, \tag{4.5}$$

the entropy function is evaluated at the interval bound closest to 0.5. If $0.5 \in [a, b]$, the maximum of entropy is $H_{max} = H(0.5)$. This is illustrated in Fig. 4.2b.

Having determined the maximum of entropy in each cell, the procedure continues as explained in Sect. 3.4.

Chapter 5

Experiments

5.1 Hardware

(a) omniRob, equipped to explore laboratory environments: Eight ToF-Cameras as well as two laserScanners and one Xtion are mounted to the robot

(b) KR16, equipped to scan objects: a laser striper is mounted

Figure 5.1: Hardware used for the simulated exploration experiments (Fig. 5.1a) and for 3D-modelling of real objects (Fig. 5.1b)

At the RMC at DLR in Oberpfaffenhofen, Germany, several robotic platforms are available for testing autonomous exploration and mapping approaches. Two of them are especially important to this thesis: A mobile platform called omniRob and the KR16 [KUKb], a large moveable arm anchored to the ground. Both are manufactured by KUKA Robotics (KUKA).
The omniRob is equipped with two S300 laser scanners, located at opposite corners of the platform. They can be used to record a two dimensional, horizontal profile of the robot's surroundings. Built-in odometry sensors determine the distance covered. A timestamp is added to both the odometry and laser scanner measurements. At DLR, the omniRob is

Table 5.1: Sensor types and specifications of the robots and sensors used in this thesis[O3D],[Sca],[Sic10]

robot type	omniRob		KR16
sensor type	O3D-100	Sick S300	ScanControl 2700-100
number of sensors	8	2	1
resolution	$64px \times 48px$	$30mm - 70mm$, configurable	$640px$
opening angle	$30° \times 40°$	$270°$	$14°$

additionally equipped with eight time-of-flight-cameras(ToF-Cameras), mounted around the robot at knee-level. They contain a light source and evaluate the time needed by the light to reach an obstacle, which results in a three-dimensional depth image. These cameras enable the robot to perceive its environment in almost any direction. As specified in Tab. 5.1, the ToF cameras of type O3D-100 [O3D] have a resolution of 64×48 pixels and an opening angle of $30° \times 40°$. Due to the number and arrangement of sensors, the omniRob is unable to perceive its entire surroundings in one scan. However, considering that the robot can move omnidirectionally -hence the name- it is able to turn around without changing its location and can therefore iteratively scan areas that are not covered by a single measurement.
For 3D-scanning purposes, the KUKA KR16 can be equipped with a ScanControl 2700-100 [Sca] laser striper. In contrast to the S300 Laser scanner, which measures time-of-flight of a laser beam, this laser striper measures object distance through triangulation. By moving the laser striper perpendicularly to the measured profile, a three-dimensional image can be constructed. Since the robotic arm has six axes, it is well-suited for modelling objects by moving around the object and gradually assembling a model from the obtained 3D-images. In this application, an exploration map is not created. Rather, a collision space is constructed around all occupied voxels and paths are planned outside of that collision space. Moreover, a frontier is not considered, which inhibits sampling scan positions. Instead, scan paths are sampled on a sphere around the entire object, regardless of which areas are known or unknown. The decision process then decides between those scan paths instead of deciding between samples around a frontier.

5.2 Simulation

Before putting all the steps above into practice, they are tested and evaluated in a simulated environment. In the following, the parts that are most important for this thesis are explained.

(a) Simulation of omniRob in laboratory with reflecting closet

(b) Bird's eye view of the room used for the exploration experiments. The closet which will have different surface properties is highlighted in red.

5.2.1 Simulating the Environment

In order to be able to compare real and simulated measurements, the robotic laboratory is imitated as closely as possible. Chairs and desks as well as doors and walls are in the same place as they are in the real laboratory. On the one hand, the 3D-model is a replica of the real-world laboratory, but on the other hand, it can be modified to simulate different environmental conditions such as less than ideal surfaces or dynamic environments.

5.2.2 Modifications to the Simulated Environment

Imitating a real-world measurement requires imitating the real-world surface properties as well as the sensor's capabilities and flaws. The following sections describe how this is implemented in the simulation.

Creating a Challenging Environment

As described above, the simulated laboratory can easily be modified for testing. In the scope of this thesis, elements with unusual surface properties are added to the map. E.g., doors are assigned reflective properties and a cupboard is given an absorbing surface. In reality, unusual surface properties such as reflective, absorbing or even transparent

surfaces are a major cause of sensor noise and can lead to erroneous interpretation of the measurement data. By introducing these properties to the simulated environment, one can safely examine strategies to cope with inconsistent measurement data.
Surface properties are assigned to both the internal storage structure of the polygons as well as to the 3D-model of the laboratory. In doing so, any interaction of the sensors with an unusual surface can be monitored in the visualization. Results can therefore be interpreted and reviewed more easily.

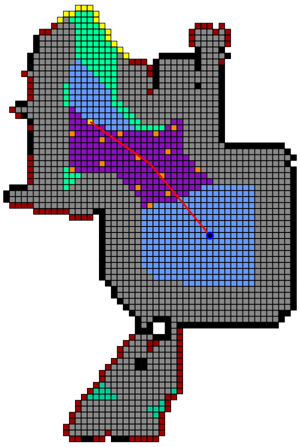
Simulating Noise

After introducing surface properties to the simulated environment, the sensors' behavior needs to be modeled. This is achieved through a function that adds noise to every distance value that is obtained from a measurement. For each measurement beam, the material of the surface intersecting with the beam is identified. Depending on the surface properties, noise is added differently.
In the ideal case, the surface is neither strongly specularly reflective nor strongly absorbing. Rather, a diffuse reflection yields best results. In that case, Gaussian noise is added. If the surface is absorbing, it is treated as if no beam is returned. Therefore, the maximal measurable distance is returned. Thus, the robot remains ignorant about the area that comprehends absorbing items. In case of a specular surface, an entirely random value is returned. Lastly, if the surface is transparent, it is randomly decided whether the surface is treated as diffuse or transparent. In case it is treated as transparent, the beam's path is followed until it reaches another surface. The procedure as well as the results of comparing the Interval Update and the Bayes Update, as well as comparing different next-best-view criteria, are explained in the following.

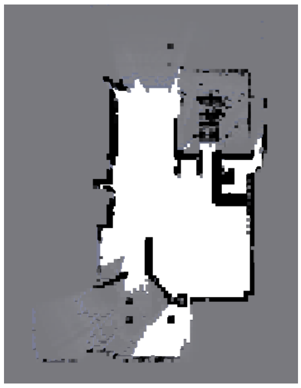
5.3 Interval vs. Bayes

The two update types that are used in this experiment have been explained in subsection 3.3.2 and Sect. 4.2. The Bayes Update has only one configuration, whereas the Interval Update is tested in three different configurations, depending on how the state of each cell is calculated. Either the upper interval bound, the lower interval bound or the average of the two bounds is defined as the state.
All of the four configurations described above are confronted with different surface types:

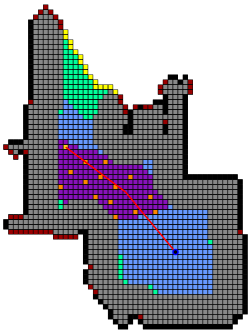
- a noise-free, ideal surface
- a reflecting surface, such as a mirror
- an absorbing surface
- a transparent surface



(a) Interval: Exploration map after initial look-around in room with ideal surfaces



(b) Interval: Voxel space after initial look-around



(c) Bayes: Exploration map after initial look-around in room with ideal surfaces

(d) Bayes: Voxel space after initial look-around

Figure 5.3: Exploration maps and voxel spaces after initial look-around using the Interval Update (top) and Bayes Update (bottom)

As a reference, an ideal surface is examined in the experiment. This means that the surface has the same properties as any other surface in the room, i.e. it returns all measurement beams to the sensor with no modification. By looking at the voxel space, it can be seen in Fig. 5.3b that the sensors detect the surface and the probability of occupancy of the surface voxels is already high after the initial look-around. The area in front of the closet is identified as free. Thus, the surface is displayed in the exploration map in Fig. 5.3a as a black line of voxels.
The voxels in the top left corner remain unknown, even though they might technically be visible from the robot's position. However, their distance to the robot exceeds the sensors' maximum measurement distance.
Moreover, the area in the lower left corner is noteworthy. Surfaces have been identified, but the area in front of the surfaces is seemingly still unknown. This is due to the fact

that the barriers that guide the robot to the upper left part of the room are fairly low. They do block part of the sensors' field of view, but allow to measure an area that is further away from the barriers. Thereby, the wall can be detected, while the area near the floor and close to the barriers remains unperceived.
Since the robot is able to definitively locate the walls that surround it, it can continue exploring the room. After about six to seven exploration steps, the reachable area is fully explored and the exploration is considered to have finished successfully. The resulting exploration map is dominantly coloured black, gray and blue, i.e. all voxels are either reachable and explored or not reachable.

The Bayes-update achieves similar results. The room is fully explored after five to six exploration steps. This means the Bayes Update is a little bit faster on average. After the initial look-around, there remain uncertain unreachable voxels in the exploration maps in both algorithms. However, their number has decreased significantly after the exploration has finished.

5.3.1 Absorbing Surfaces

(a) Voxel space after initial look-around

(b) Voxel space after the outline of the surface has become visible in spite of its absorbing property

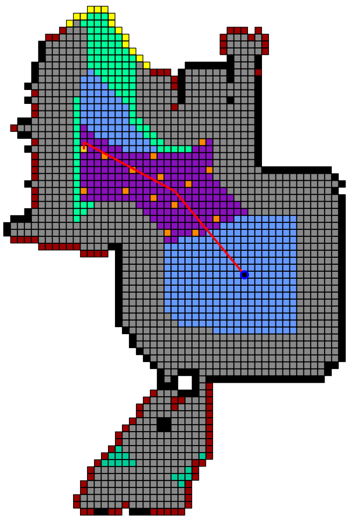
Figure 5.4: Voxel space after initial look-around (Fig. 5.4a) in comparison to the space after the outline of the surface has been identified (Fig. 5.4b).

The first non-ideal surface that is examined in the experiment is absorbing, which means that some of the measurement beams are not returned, resulting in no measurement data. It can be seen in Fig. 5.3a that the sensors do not receive much data from the surface of the closet, thus the area remains unknown. However, the voxel space (Fig. 5.4a) already reveals that some measurement beams are returned by the surface. The front of the closet is visible as a series of black voxels, while the unreturned measurement beams have led to an accumulation of gray, i.e. unknown voxels in front of the closet. Since the area in front of the surface is still unknown, the shape of the absorbing object can not yet be displayed in the map.
Moreover, when examining the interval width and the state simultaneously, the underlying process becomes clearer. After the initial look-around, the certainty about the voxels along the walls is quite high, therefore they are coloured black. Since hardly any measurement beams are returned by the absorbing surface, the uncertainty of the voxels in front of the critical surface remains high. Furthermore, an approximate outline of the surface itself can be identified.
Due to the fact that border of the unknown area is reachable, the algorithm selects the border as the new frontier and continues measurement of the region. In spite of most beams being absorbed, occasionally a beam is reflected to the sensor. By viewing the absorbing surface repeatedly, the algorithm collects data from the measurement beams that did return and is eventually able to reconstruct the surface, as seen in Fig. 5.3b and Fig. 5.4a This is achieved after the second exploration step. The robot can then continue to further explore the room and is able to complete the exploration.

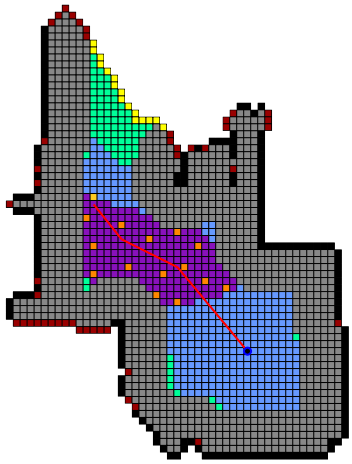
The Bayes Update performs similarly well. The exploration maps after the initial look-around and after the first exploration step are shown in Fig. 5.3c and Fig. 5.3d. In the first step, both algorithms tackle the absorbing surface, examining it more closely. Since some measurement beams are returned correctly, the surface outline is determined correctly after one or two exploration steps. The robot then continues to explore the room. When comparing the voxel spaces of the Bayes Update and the Interval Update after the first exploration step, it becomes evident that the Bayes Update updates the state faster, i.e. in fewer exploration steps. On average, it takes one exploration step for the Bayes Update to correctly detect the outline of the aborbing surface, while it takes two to three steps for the Interval Update. The exploration map in Fig. 5.3c shows that the Bayes Update has already identified most of the voxels in front of the absorbing surface as free just after the initial look-around. The reason behind this slower update has been explained in Sect. 4.2.

5.3.2 Transparent Surfaces

Another non-ideal surface that is examined in the experiment is transparent, which means that some of the measurement beams are reflected by the object's surface, while other beams pass through the object and are reflected by the surface behind the object. This results in two conflicting measured distances, both of which can be supported with data.



(a) Exploration map after initial look-around in room with transparent surface



(b) Bayes: The shape of the closet can be seen clearly in spite of its transparent property

Figure 5.5: Exploration maps after initial look-around. The left image is generated using an interval update with the mean of the interval as the state, while the right image is generated using a Bayes Update.

However, it can be seen in Fig. 5.5 that the transparency does not result in an erroneous map representation. Only when examining the voxel space, the effect of the transparent surface becomes apparent, as the wall behind the cupboard is visible as well. Again, the Bayes Update performs similarly well. A mostly noise-free representation of the transparent surface is visible in the exploration map immediately after the initial turn-around and the sensor noise does therefore not pose an obstacle to the exploration.
The fact that both update types perform well in this experiment regardless of the transparency is owed mainly to two circumstances:

- The surface of the closet facing the robot is a flat plane apart from some small details like handles of the cabinet doors. This allows the surface to be easily fitted into the voxel space grid.
- Moreover, the transparent closet is placed right in front of a wall. Even if the light beams pass through the closet, they are reflected by the wall. Thus, only the closet and the wall behind it are measured when aiming the sensor's beams at it. If the closet was placed in the middle of the room, light passing through the closet would be reflected by other objects in the room. If the closet was viewed at a different angle, light would be reflected from different objects. Instead of measuring two different distances, the sensors would measure a wide range of distances, which would be more difficult to interpret.

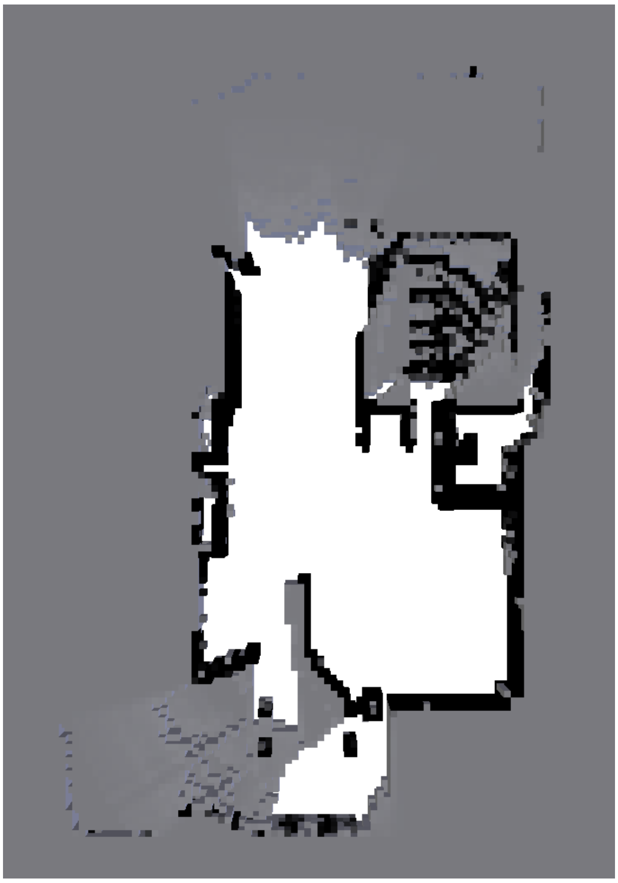
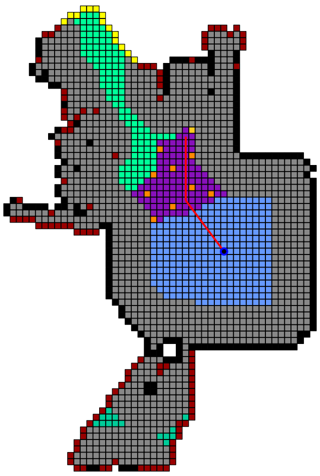


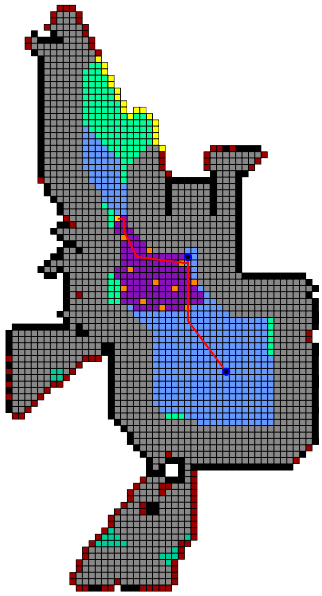
Figure 5.6: Voxel space after initial look-around in room containing a transparent surface

5.3.3 Reflecting Surfaces

The third non-ideal surface that is examined in the experiment is reflecting, which means a random distance value is returned if the measurement beam hits the surface. As a result, all voxels in the area in front of and behind the surface are updated with conflicting data. Some free voxels are set an occupied state, while the state of voxels that contain an actual obstacle remains unknown or is set to free. Both the voxel space and the exploration map reflect the erroneous data, and the surface of the closet can not be identified in the voxel space nor in the exploration map.
The voxels that have spuriously been defined as occupied are interpreted as obstacles in the exploration map and thus restrain the robot from continuing the exploration through other parts of the room. In some cases, however, the erroneous voxels are located in a way that lets the robot pass the closet and continue exploration.
The Bayes Update copes similarly with the reflecting surface. Some voxels are spuriously set to an occupied state, while others remain unknown or are set to a free state. As a result, the outline of the reflecting surface is occluded by voxels with an erroneous state. These occupied and unknown voxels are spread across the left half of the room, such that

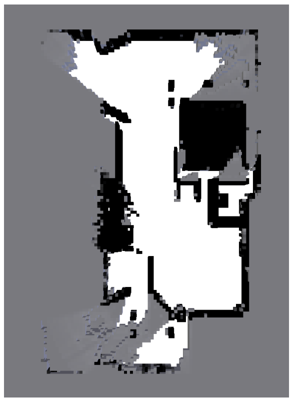


(a) Exploration map after initial look-around in room with reflecting surface



(b) The exploration map proves that the robot is able to continue exploration past the reflecting closet

(c) Voxel space after initial look-around

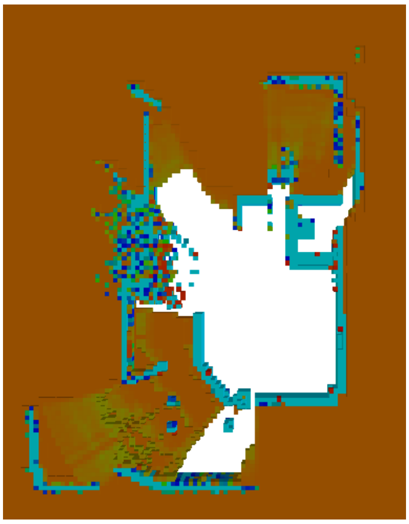


(d) The robot successfully continues exploring the unknown space in spite of the presence of a reflecting surface

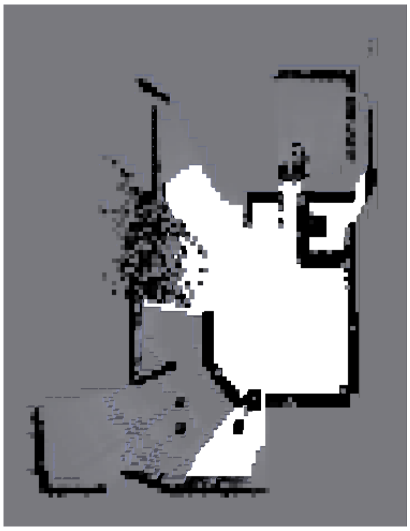
Figure 5.7: Exploration map and voxel space after initial look-around and after the reflecting surface has been passed. The simulation uses interval update with the mean of the interval as the state

the robot can not pass the reflecting cupboard. It can thus not continue exploration. It can therefore be stated that the qualitative performance of the Bayes Update is worse than that of the Interval Update with regard to this simulated reflecting surface.

The voxel space can be visualized differently to illustrate how the update process influences the interval width. A possible visualization is depicted in Fig. 5.8. The figure depicts a



(a) Color coded representation of interval bounds: Blue: narrow interval, Red: Broad interval



(b) Grayscale representation of voxels' states

Figure 5.8: Interval width and probability of occupancy in a voxel space

voxel space after an initial look-around in a room with a reflecting surface. In the image on the right hand side, the nuances of gray represent the state of the voxel $Pr(Occ)$. The gray scale is identical to the one explained in subsection 3.3.2. The image on the left illustrates the interval width of the voxels. Blue voxels contain a narrow interval, while red voxels contain a broad interval. The majority of voxels is orange in the color-coded map, since the inital interval is set to [0.05; 0.95]. The interval can broaden under certain circumstances, creating red voxels. Usually the interval narrows over time, creating ochre, green, turquoise and blue voxels.

In Fig. 5.8, the main source of conflicting data is the reflecting surface on the left side of the room. Thus, detecting conflicting measurements could be implemented by analyzing the interval width as well as the state.

In Fig. 5.9, a combination of interval width and state value is visualized. Tab. 5.2 describes which colors represent which state and which interval width. A broad interval is coloured red, while narrow intervals are coloured depending on their state. Gray voxels are neither definitively occupied or free, but the state has considerably shifted from its initial value of 127. Again, black voxels are treated as obstacles, while uncertainty remains about blue and gray voxels. It can be seen that obstacles are assumed where walls are and most free area is certain to be free. Furthermore, the majority of gray and yellow voxels appears within the area in front of the reflecting surface. This phenomenon can be explained as follows: Since the reflecting surface leads to a measurement of nearly random distance values, each cell within a certain distance in front of and behind the surface is measured to have random

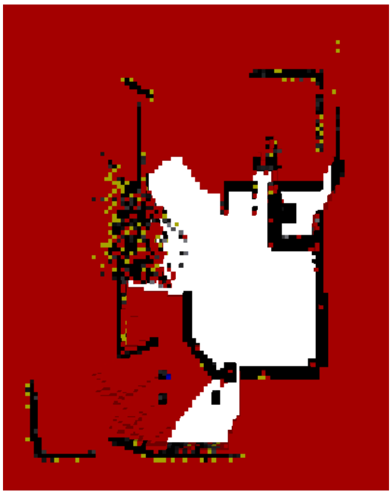


Figure 5.9: Combination of interval width and state

Figure 5.10: Detail of 5.9

Color		Interval Width	State
red		>127	any
blue		≤127	75<state<180
black		≤127	>250
white		≤127	≤4
gray		≤50	all other states
yellow		50≤width≤127	all other states

Table 5.2: Color code of Fig. 5.9 and Fig. 5.10. Each color represents a different combination of state and interval width

probabilities of occupancy. It is unlikely that the number of measurements in which a voxel is supposedly free will equal the number of measurements in which a voxel is supposedly occupied, such that they would cancel out. Therefore, the probability of occupancy of most voxels shifts away from the average 127. Furthermore, it is also unlikely that the voxel is measured to be free each time, just as it is unlikely to be measured as occupied each time. However, once one of the interval boundaries reaches either the upper or the lower numerical limit, i.e. 255 or 0, it is numerically impossible for that bound to change to another value. Thus, a few consecutive similar measurements are sufficient to define a voxel as free or occupied, which explains the amount of black voxels in the area.

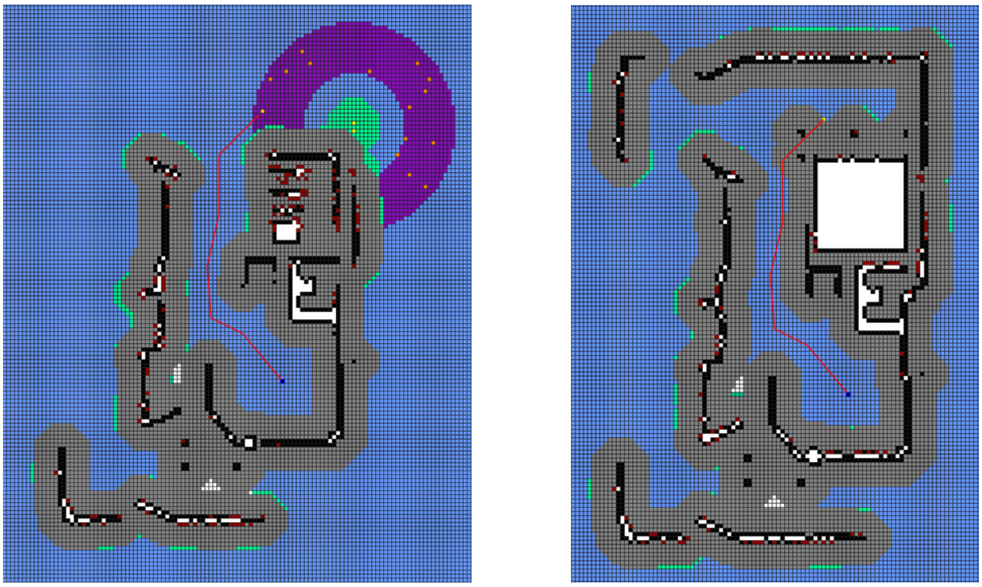
5.3.4 Examination of State Definitions

As explained in Sect. 4.1, the state can be defined in different ways when using the Interval Update. Experiments are conducted for the three state definitions as explained at the beginning of Sect. 5.3. The effects of taking the mean of the interval as the state have

If voxels are black in the grayscale map and blue or turquoise in the color-coded map, they can be interpreted as areas that have repeatedly been measured as occupied. Therefore, they are usually actual obstacles.

Voxels that are gray on the grayscale map and orange on the color-coded map have usually not been perceived by the robot.

The voxels that remain are mostly blue and gray, which means they contain a narrow interval, but their state is still "unknown". These are voxels that the robot has scanned and updated repeatedly, but which have become neither free nor occupied in spite of the update steps. This means they have been updated with conflicting measurements.



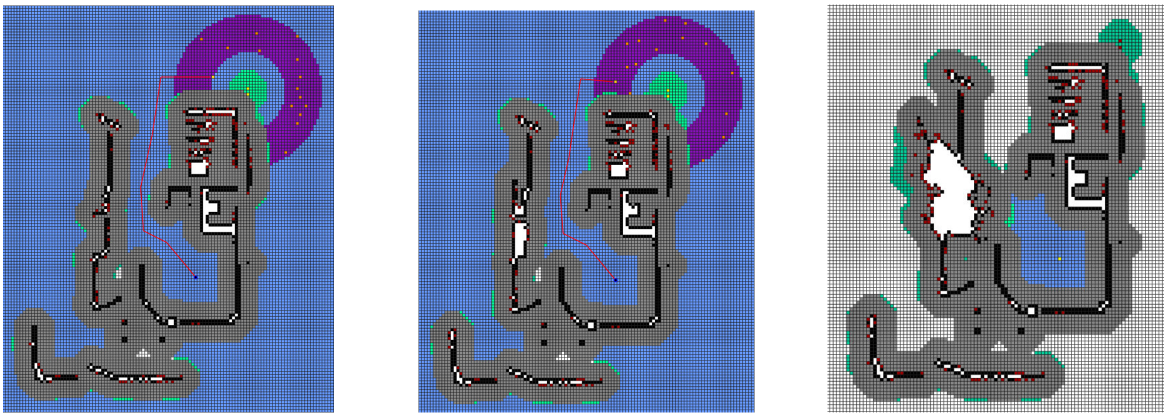
(a) Exploration map after initial look-around using the lower interval bound as the state.

(b) Exploration map after second, final exploration step.

Figure 5.11: Exploration map after first and second exploration step.

already been examined above. In the following, the effects of the other two state definitions will be explained.

Initially, the interval always ranges from about 0.05 to about 0.95. Thus, if the state is defined as the lower interval bound, the entire map is initially set to have a "nearly free" state of 0.05. The entire voxel space as well as the exploration map is free, unless the robot measures an obstacle. After the initial look-around, the robot has measured some



(a) Absorbing surface

(b) Transparent surface

(c) Reflecting surface. The closet is impassable due to its surface properties.

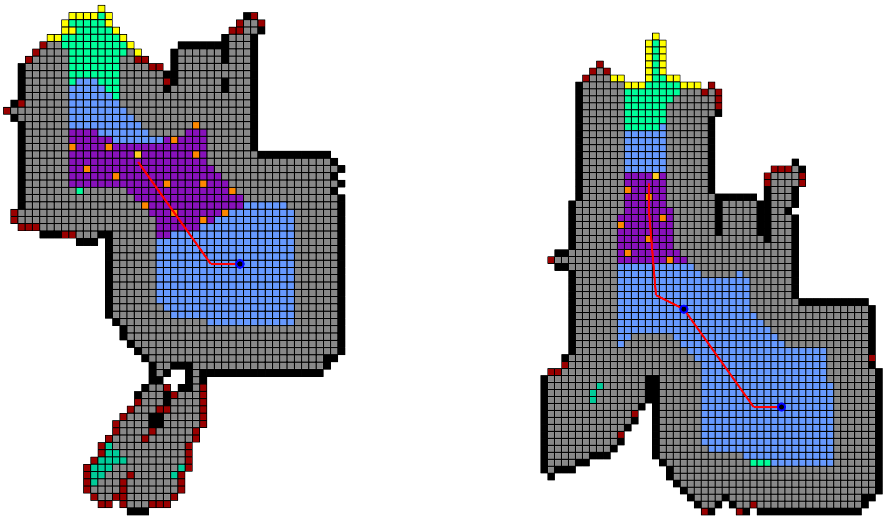
Figure 5.12: Exploration map of rooms with different non-ideal surfaces using the lower interval bound as the state.

obstacles close to its position. However, the area behind the obstacles remains free, just like it remains unknown if the mean is chosen as the state. Therefore, the frontier between areas that have and those that have not been scanned is usually not visible. In these cases, the algorithm chooses any other reachable unknown voxel as the frontier. In this setting, unknown voxels often have been perceived and updated towards an occupied state, but have not reached that state yet. The robot then performs its next-best-view planning on that frontier and moves to the selected position.

Fig. 5.11a shows an exemplary exploration map. It can be seen that almost any voxel which is not an obstacle or a collision space is both free and reachable. Thus, the robot concludes after few exploration steps that all reachable voxels are known, i.e. free or occupied. The exploration is terminated. The resulting exploration map is shown in Fig. 5.11b. There are no red or yellow voxels left, i.e. no unknown reachable voxels are present in the map. This is largely due to originally setting all voxels to a defined state.

Figures Fig. 5.12a-Fig. 5.12c show the exploration map of the same room, except that the closet on the left hand side of the room has different surface properties in each of the images. It can be seen that the algorithm copes well with both the absorbing surface and the transparent surface, and the robot can continue the exploration in both cases. The reflecting surface, however, is critical, since the algorithm identifies it as an accumulation of unknown voxels, which are bordered by a few occupied voxels, making the area unreachable. The robot cannot continue with exploration and terminates the process.

The same experiments are conducted with the state defined as the upper interval bound. Thus the entire map's probability of occupancy is initially set to about 0.95. The entire voxel space as well as the exploration map is occupied, unless the robot measures a free area. After the initial look-around, the robot has measured some free space around itself and is able to move away from its original location. The voxel spaces generated from these



(a) Exploration map after initial look-around using the lower interval bound as the state.

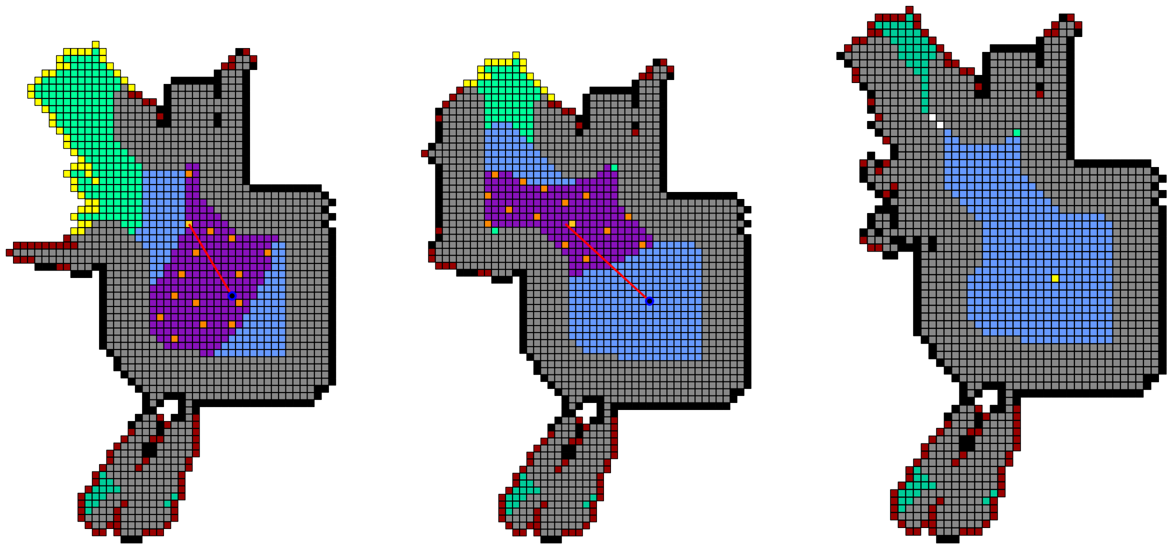
(b) Exploration map after second, final exploration step.

Figure 5.13: Exploration map after first and second exploration step, using the lower interval bound as the state in a room with ideal surfaces

experiments are shown in Fig. 5.14d - Fig. 5.14f. It can be seen that the results on the absorbing and transparent surface are similar to the ones of the Interval Update with the mean as the state. However, the results in the room with the reflecting surface are worse. This is likely due to the fact that once a state reaches the numerical limit, i.e. 0 or 255, an update has no effect on it any more. This has been explained in subsection 3.3.2. Thus, if the voxels with a state of “nearly occupied” are spuriously measured as occupied, the state is likely to reach the numerical limit. This could be the source of the accumulation of occupied voxels in front of the reflecting surface.
In contrast to the configuration that uses the lower interval bound as the state, frontiers are defined similarly as in the Bayes Update and the mean configuration of the Interval Update. The voxels previously set to occupied are measured as free and thus their state is lowered, becoming unknown and eventually free. The border between those unknown and free voxels is set as the frontier. The results are therefore similar to the ones from the mean configuration of the Interval Update.

5.4 Maximum Entropy vs. Entropy

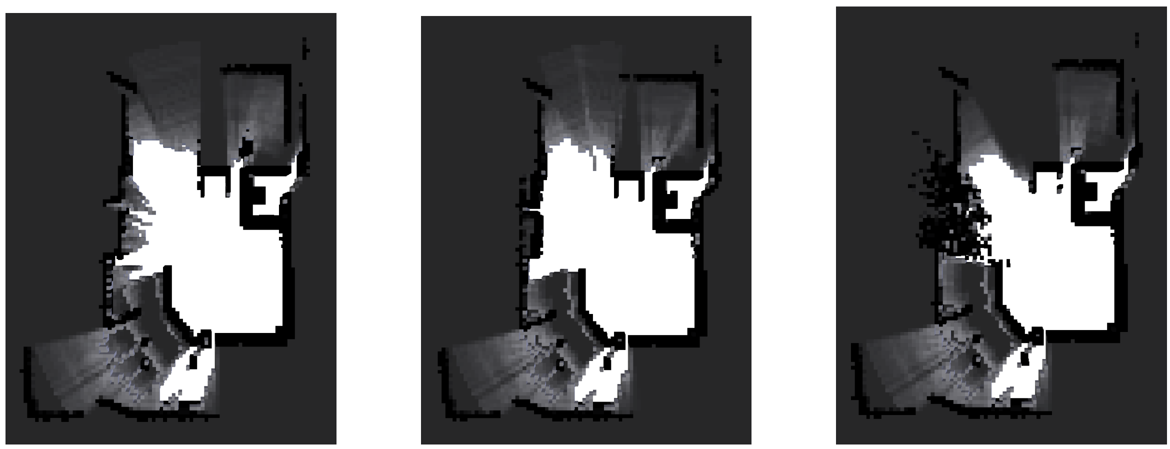
After these tests have been conducted, next-best-view criteria are evaluated. The so-called Entropy Criterion is explained in Sect. 3.4, while the criterion based on the maximum of entropy is introduced in subsection 4.3.2. Since both next-best-view criteria are determined



(a) Absorbing surface: Exploration map after initial look-around using the lower interval bound as the state.

(b) Transparent surface: Exploration map after initial look-around. The transparent surface does not hinder exploration.

(c) Reflecting surface: Exploration map after initial look-around. The exploration is terminated, since no frontiers are found.



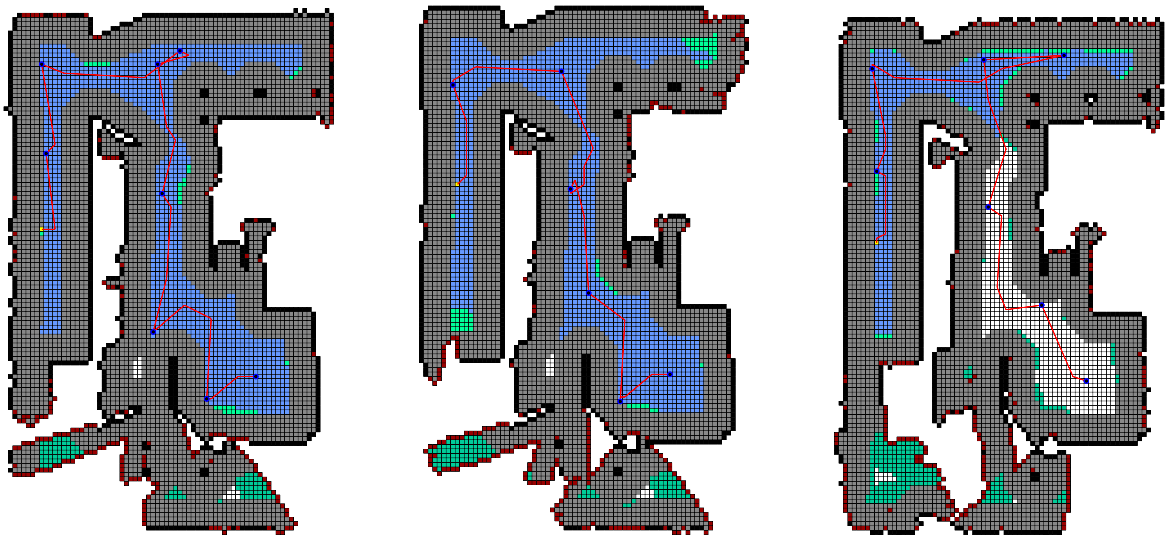
(d) Absorbing surface: The outline of the surface is visible in the voxel space

(e) Transparent surface: Both the surface outline and the wall behind the closet are visible

(f) Reflecting surface: The measurements are too noisy to allow visibility of the closet

Figure 5.14: Exploration map and voxel space of rooms with different non-ideal surfaces using the upper interval bound as the state.

similarly, identical calculations can be necessary for both criteria in some special cases. If the upper interval bound lies below 0.5, the Maximum Entropy Criterion uses the upper interval boundary for its entropy calculation. If the state is defined as the upper interval bound, the Entropy Criterion effectively uses the same values and yields the same results.



(a) Entropy Criterion: Exploration is finished after eight exploration steps

(b) Maximum Entropy Criterion: Exploration terminates after six exploration steps

(c) Bayes with Entropy Criterion: Exploration terminates after seven exploration steps

Figure 5.15: Exploration in a room with an absorbing surface. In this case, the Maximum Entropy Criterion needs the fewest exploration steps.

Similarly, if the lower interval bound exceeds $Pr(Occ) = 0.5$ and the state is defined as the lower interval bound, the calculations of both criteria are identical.
As the name implies, a NBV criterion only affects the selection of NBV positions. Thus, it determines which voxels are perceived and thus updated. In combination with the update type, which determines how the collected information about the affected voxels is combined, i.e. how the are updated, it can result in a different update process.
The results from experiments in a room with an absorbing surface are depicted in Fig. 5.15. Fig. 5.15a displays the exploration map after the final exploration step with an Interval Update. NBVs have been determined using the Entropy Criterion. Fig. 5.15b has been generated under similar conditions, except that the Maximum Entropy Criterion has been used to determine NBVs instead. Fig. 5.15c shows the exploration map after exploration with the Bayes Update has terminated. As before, the Entropy Criterion is used to determine NBVs. Comparing the results from exploration with both criteria, it can be seen that an exploration process using Maximum Entropy Criterion terminates faster. In some cases, as few as six exploration steps are needed when coping with an absorbing surface. About seven exploration steps are needed on average. When exploring a room with a reflecting or transparent surface, an average of seven exploration steps are needed to explore the entire room. Again, exploration with the same update type but using the Entropy Criterion needs more steps on average. The Bayes Update, however, also needs about seven exploration steps to explore the entire room if the robot is able to pass the surface causing depth noise. Thus, planning NBVs with the Maximum Entropy Criterion is approximately as fast as an

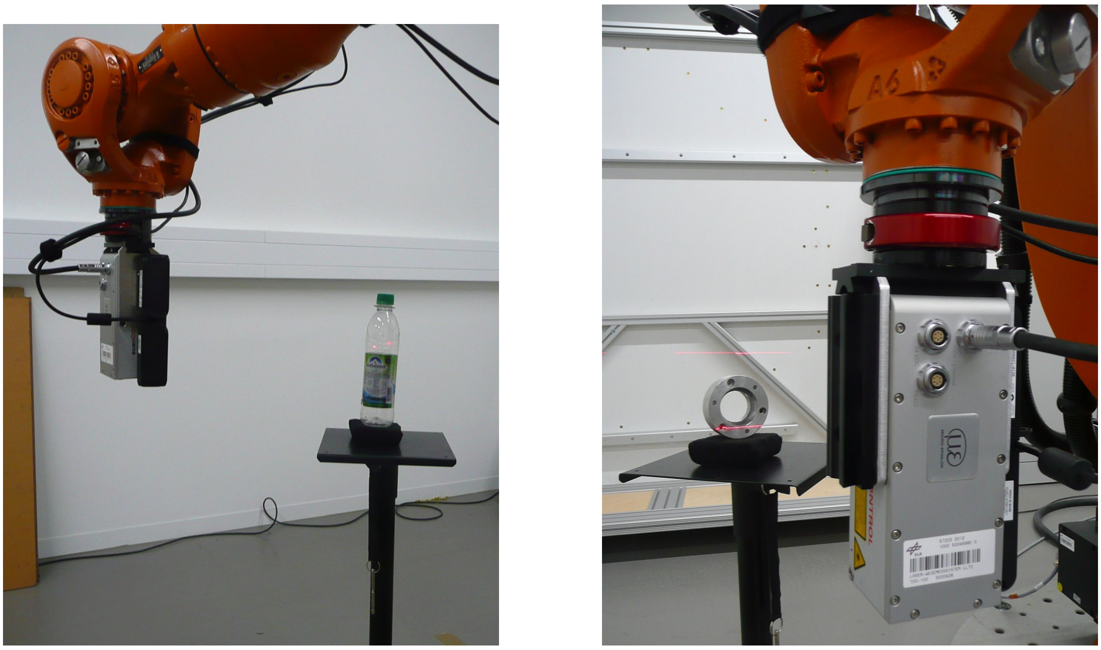
exploration with the Bayes Update. In this simulation, the NBV Criterion compensates the additional time needed by the Interval Update.

5.5 Tests on Hardware

In order to verify the results obtained from experiments in the simulated environment and to prove the portability of the concept, experiments are conducted on hardware. A robotic arm, the KR16 introduced in Sect. 5.1, is used for autonomous object modelling. The space update type as well as the next-best-view criterion are transferred to the implementation of the autonomous scanning algorithm. The Bayes Update is used as a reference. Reflecting, transparent and light-absorbing objects are tested. Not all results will be transferable to this application. In object modelling, the robot needs to be able to scan the object from each direction. This inhibits a setting like in the exploration experiments. Thus, a transparent object, for instance, can not be placed in front of a perfectly diffusely reflecting surface. The performance will likely be worse.

Fig. 5.16 shows the KR16 robotic arm while scanning objects. The reflections of the red laser can be seen on both the mechanical component (Fig. 5.16b) and the transparent bottle (Fig. 5.16a). The mechanical part reflects light in various directions. In contrast, the transparent object reflects hardly any light. As seen in the photograph, light is reflected towards the camera at two points due to the embossing of the bottle. Most of the light passes through the object. Moreover, when looking at the mechanical component, it can be seen that a 2D profile of the object is determined at once. The laser line is visible on the component itself as well as on the wall behind the object. By moving the sensor perpendicularly to that line, a 3D depth image is obtained. A voxel space and a mesh are then constructed from the depth image.

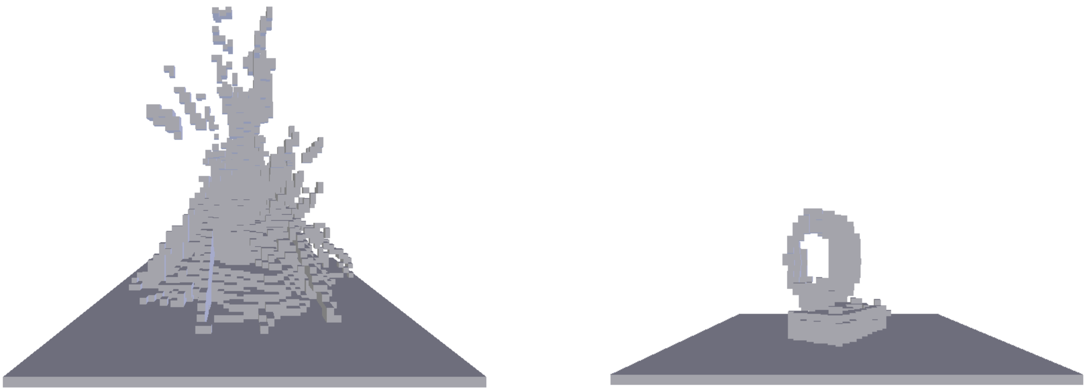
Fig. 5.17 shows the results of scanning the mechanical component. One of the scans has been performed according to the state-of-the-art, updating the voxel space with a Bayes Update. The other one has been obtained in a similar way, except that an Interval Update has been used to update the voxel space. When using the Bayes Update, the shape of the object cannot be distinguished, since it is covered up by other voxels that have spuriously been measured as occupied. In contrast, when using the Interval Update, the outline of the object can easily be determined. The circular shape as well as the hole in the objects center are visible. However, this volumetric model is not a perfect representation of the object either. Some voxels have been measured as free, even though they lie within the object. Thus, the outline of the object could become indistinct, or the robot could collide with the object in a worst case scenario. Nevertheless, judging by the voxel space, the Interval Update seems to cope better with the noise from a reflecting surface. This could be due to difference in speed between Bayes and Interval Update. While the Interval Update is slower, leaving voxels unknown for a longer amount of time, the Bayes Update is faster, thereby prematurely giving voxels an occupied state.



(a) A transparent water bottle is scanned

(b) A mechanical component with a reflecting surface

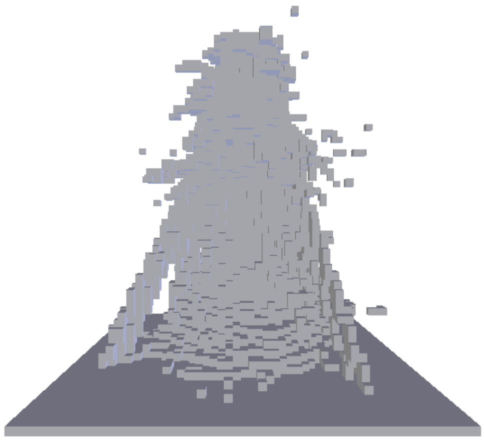
Figure 5.16: Objects with different surface properties are scanned with a laser striper mounted to a KR16 robot



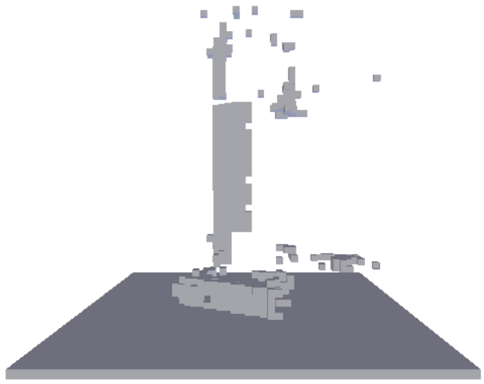
(a) Voxel space resulting from scanning a mechanical component using the Bayes Update

(b) Voxel space resulting from scanning the same object using an Interval Update

Figure 5.17: Voxel space resulting from scanning a mechanical component with a laser striper mounted to a KR16 robot.



(a) Voxel space resulting from scanning a transparent bottle using the Bayes Update



(b) Voxel space resulting from scanning the same object using an Interval Update

Figure 5.18: Voxel space resulting from scanning a transparent bottle with a laser striper mounted to a KR16 robot.

When examining the transparent object, the feature of the Interval Update is not as advantageous. The results from updating the space with the Bayes Update and with the Interval Update are shown in Fig. 5.18. The majority of the bottle is perceived as free when updating the space with the Interval Update during the scan. The outline of the object is indistinct, since only one side of the object is seen in the voxel space. The Bayes Update, however, does not perform much better, leaving the shape of the bottle obscured by voxels with an erroneous state. The result of the scans using different update types are comparable, since neither achieves satisfactory results.
The following can be concluded from these results: Both the Bayes Update and the Interval Update are error prone when confronted with real surfaces that do not reflect diffusely. The Bayes Update tends to assume too many occupied voxels, while the Interval Update tends to assume too much free space. Depending on the surface properties, the behavior is less or more distinct. The Bayes Update is extremely error prone when scanning a strongly reflecting object, and copes better with the transparent object. The Interval Update is more prone to errors with transparent objects, and performs fairly well with a reflecting object. The performance of the two update types is comparable when confronted with transparent objects, while the Interval Update performs considerably better when confronted with a reflecting surface. Thus, it is concluded that the overall performance of the Interval Update is better. However, when scanning fragile objects or in safety critical applications, the behaviour of the update should be considered. An additional safety distance could be introduced to avoid collisions with obstacles within the wrongly assigned free space.

5.6 Summary of Experiments

Experiments have been conducted to examine four types of a space update: A state-of-the-art Bayes Update and an Interval Update with three different state definitions. The experiments comparing different state definitions have shown that the state influences the performance of the update. Defining the state as the lower interval bound is a very optimistic approach, while choosing the upper interval bound as the state yields a more careful exploration. In future work, other state definitions that lie between the definitions examined in this thesis could be trialled.

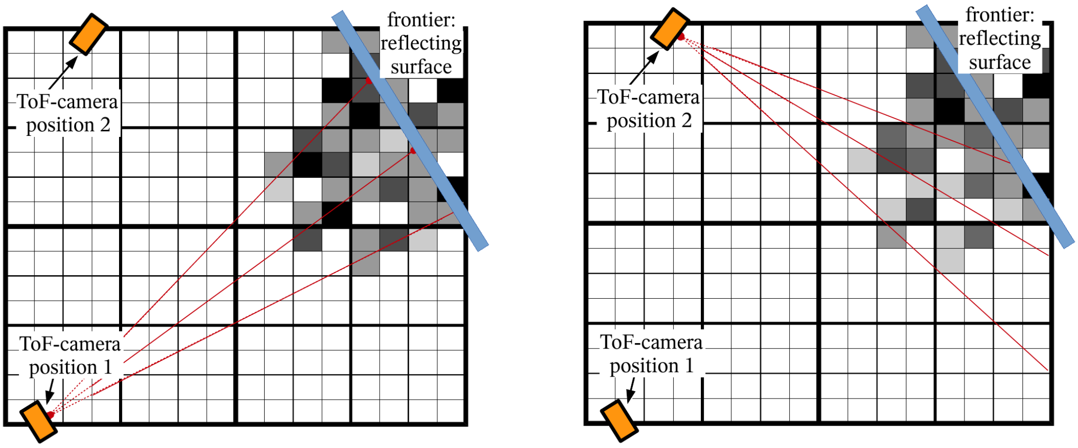
Moreover, the simluated experiments showed that Interval Update needs more exploration steps on average. However, the number of required exploration steps has been reduced by using another NBV criterion, namely the Maximum Entropy Criterion. Hence, the number of exploration steps that are needed on average for an Interval Update with a Maximum Entropy Criterion are about equal to the number of steps needed for exploration with the Bayes Update and an Entropy Criterion.

The results from the simulated experiments and the experiments on hardware are generally comparable. In both cases, the two update types perform similarly well on most surfaces. The conclusion from the simulated experiments suggested that the Interval Update performs better on reflecting surfaces. The same conclusion can be drawn from the hardware experiments described above. In both the simulated and the hardware experiments, the Interval Update interprets more space as free than the Bayes Update. Considering the limitations of the experiments, e.g. sample size, possible errors in simulating noise and possible bias by the test supervisor, it can be deducted that the Interval Update is advantageous in coping with reflecting surfaces. Taking into account that, on other surfaces, its performance is comparable to that of the Bayes Update, it can be said that the update is advantageous when dealing with conflicting data in general. However, this needs to be proven by additional experiments in the future.

Chapter 6

Conclusion and Outlook

In this thesis, a state-of-the-art exploration algorithm has been extended to better cope with sensor noise from non-ideal environments. The interpretation of data and, more specifically, the update of a space representation based on that data has been identified as a good starting point to tackle the issue. Thus, a novel update type based on the concept of imprecise probabilities, the so-called Interval update, has been introduced. Instead of considering only a probability of occupancy for each element in the space, it adds a measure of uncertainty to each voxel. This allows for a distinction between ignorance and uncertainty about elements of the space that is to be explored. The Interval update has successfully been developed, implemented and tested. The tests have been conducted in a simulation, as well as on hardware. Three variants of the novel update type have been designed, each of which have been examined in the simulation. They differ in their definition of the state of the voxels, giving insight on how the state definition influences the update. It has been discovered that defining the voxel's state as the mean of the Interval is most promising, while defining the state differently can be helpful in certain situations as well. Defining the state as the lower interval bound yields a a fast, optimistic exploration. Defining the state as the upper interval bound is a rather pessimistic approach, suited if a slower, cautious exploration is desired.
The experiments have proven that, if the voxel's state is set to the median, the novel update can compete with a state-of-the-art Bayes update. When comparing the performance of the two update types, it is observed that the Interval update does need a little more time and storage space. However, the results from the simulated experiments suggest that it offers better performance when exploring environments with reflecting surfaces. The ability to distinguish between ignorance and uncertainty, i.e. to identify conflicting data, has been demonstrated based on the results of the simulation. A strategy to use the additional information to its full capacity, namely decision on NBV based on the maximum of entropy, has been suggested, implemented and tested in the simulation.
Experiments on hardware show that the update type can be applied to real data as well. Again, the performance of both update types is comparable. Moreover, the Interval



(a) First scan of the surface, generating noisy data

(b) Second scan of the area from a different angle, reducing the noise

Figure 6.1: Reducing measurement noise by scanning from different angles

update performs better on reflecting surfaces, thereby supporting the conclusion deducted from the simulated experiments.

As explained in Chap. 5, considering the interval width and the state of a voxel simultaneously can be beneficial. Whenever a voxel's state is unknown and its interval is narrow, it is considered to have been updated by conflicting measurements. If several of such voxels occur within a relatively small area, there is likely a non-ideal surface inducing noise in the measurement data. One way of identifying an accumulation of voxels with unknown state and narrow interval could be searching neighbouring voxels for similar properties.
Such an area can then be identified as problematic, and the issue can be tackled. For instance, it is advantageous to view the problematic area from a different angle, such that the sensors perceive the free area around the surface rather than the noisy surface itself.
Revising the algorithms introduced previously, the next-best-position samples are selected from a ring-shaped area around the frontier. This can be done again, except that the distance from the sampled position to the current position should be taken into account. A large distance will result in a larger difference of viewing angles, thereby reducing the noise from the problematic surface. The reduction of noise is depicted in Fig. 6.1. Note that in contrast to the statement of subsection 3.3.1, a flat scanning angle is advantageous in this case. The same matter that is an issue when detecting a perfect surface becomes a benefit when trying to approximate reasonable results from unreasonable data.
In 3D-scanning and object modelling, identifying voxels which remain uncertain could also be beneficial. If a voxel's state is unknown and the interval is narrow, i.e. if it has been updated with conflicting data, the data can be discarded, as it will likely impede correct object modelling. Thus, the majority of the remaining data will be non-conflicting, i.e. a

mesh which is constructed from that data is hoped to contain fewer flaws.
Another approach to resolve the discrepancy between conflicting datasets could be the usage of a different sensor. For instance, a 2D-camera could be used to perceive the area. However, this would require an entirely different algorithm to interpret data measured by the camera. In certain cases, heat or ultrasound sensors could also be trialled, if available on the platform.
In future work, other strategies suggested to cope with the additional information could be implemented. The areas inducing irregular depth noise could thereby be identified and the conflict could be resolved by scanning the surface with a different sensor or from a different angle. Moreover, the idea of introducing imprecision to depth-measurement interpretations could be expanded to cope with conflicting data from other measurement devices. For example, the interpretation of odometry data could benefit from this approach, if multiple odometry sensors are mounted to one device.
Overall, the idea provides many possibilities for coping with depth noise. Further investigation and implementation of the above suggestions will likely improve results. Thus, additional research in the field is recommended.

List of Figures

List of Tables

Appendix A

Abbreviations

2D two-dimensional

3D three-dimensional

DLR German Aerospace Center, Deutsches Zentrum für Luft- und Raumfahrt

KUKA KUKA Robotics

MCDM Multi-Criteria Decision Making

NBV Next-Best-View

NBS Next-Best-Scan

PEK Department of Perception and Cognition

RMC Robotics and Mechatronics Center

ToF Time-of-Flight

Bibliography

[AM01] Abellan and Moral. Building classification trees using the total uncertainty criterion. 2001.

[AM03] Abellan and Moral. Maximum of entropy in credal classification. *ISIPTA03*, 2003.

[ARS13] Nicolas Alt, Patrick Rives, and Eckehard Steinbach. Reconstruction of Transparent Objects in Unstructured Scenes with a Depth Camera. Melbourne, Australia, September 2013.

[BA09] N. Basilico and F. Amigoni. Exploration strategies based on multi-criteria-decision-making for an autonomous mobile robot. pages pp.259–264, 2009.

[CLK11] Shengyong Chen, Youfu Li, and Ngai Ming Kwok. Active vision in robotic systems: A survey of recent developments. *The international journal of Robotics Research*, 2011.

[Dem68] Arthur P. Dempster. A generalization of the bayesian inference. *Journal of Royal Statistical Society 30*, 1968.

[Dic] Oxford Dictionaries. robot. accessed October 11, 2015.

[DLR] Institute of robotics and mechatronics. accessed 23.10.2015.

[FSK09] F. Farshidi, S. Sirouspour, and T. Kirubarajan. Robust sequential view planning for object recognition using multiple cameras. *Image Vision Comput.*, 27(8):1072–1082, July 2009.

[GBL02] Hector H Gonzales-Banos and Jean-Claude Latombe. Navigation strategies for exploring indoor environments. *The International Journal of Robotics Research*, 21(10-11):829–848, 2002.

[HBAB11] Dirk Holz, Nicola Basilico, Francesco Amigoni, and Sven Behnke. A comparative evaluation of exploration strategies and heuristics to improve them. In *European Conference on Mobile Robots, ECMR 2011*, pages 25–30, 2011.

[HCK+89] M. Hebert, C. Caillas, E. Krotkov, I. S. Kweon, and T. Kanade. Terrain mapping for a roving planetary explorer. volume vol. 2, pages pp. 997–1002., May 1989.

[HHHL+14] L. Heng, D. Honegger, G. Hee Lee, L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys. Autonomous visual mapping and exploration with a micro aerial vehicle. *Journal of Field Robotics*, 31(4):654–675, 2014.

[Joy08] James Joyce. Bayes' theorem. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. The Metaphysics Research Lab, Center for the Study of Language and Information, Stanford University Stanford, CA 94305-4115, fall 2008 edition, 2008.

[Jud88] Scott M. Juds. *Photoelectric Sensors and Controls: Selection and Applications*. CRC Press, 04 1988.

[KBM+13] Simon Kriegel, Manuel Brucker, Zoltan-Csaba Marton, Tim Bodenmüller, and Michael Suppa. Combining object modeling and recognition for active scene exploration. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2384–2391, Tokyo, Japan, November 3–7, 2013. IEEE.

[KBSH11] Simon Kriegel, Tim Bodenmüller, Michael Suppa, and Gerd Hirzinger. A surface-based next-best-view approach for automate 3d model completion of unknown objects. 2011.

[KRB+12] Simon Kriegel, Christian Rink, Tim Bodenmüller, Alexander Narr, Michael Suppa, and Gerd Hirzinger. Next-best-scan planning for autonomous 3D modeling. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pages 2850–2856, Vilamoura, Algarve, Portugal, October 7–12, 2012. IEEE.

[KRBS13] Simon Kriegel, Christian Rink, Tim Bodenmüller, and Michael Suppa. Efficient next-best-scan planning for autonomous 3d surface reconstruction of unknown objects. *Journal of Real-Time Image Processing*, pages 1–21, 2013.

[KUKa] KUKA Systems Spot welding. electronic. accessed 10.11.2015.

[KUKb] Specification robots KR 6, KR 16, KR 16 L6, KR 16 S. electronic. accessed 12.10.2015.

[MHH12] Ouk Choi Miles Hansard, Seungkyu Lee and Radu Horaud. *Time of Flight Cameras: Principles, Methods, and Applications*. Springer Briefs in Computer Science, 2012.

[MLD07] Martin Magnusson, Achim Lilienthal, and Tom Duckett. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10):803–827, 2007.

[O3D] O3D100, PMD 3D sensor. electronic. accessed 13.7.2015.

[Sca] scanCONTROL 2700: Compact 2D/3D profile sensor with integrated controller. electronic.

[SH07] M. Suppa and G. Hirzinger. Multisensory exploration of robot workspaces. *Tm - Technisches Messen*, 74:139–146, 2007.

[Sha76] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, 1976.

[Sha92] Glenn Shafer. *The Dempster-Shafer theory*, pages 330–331. Wiley, 2 edition, 1992.

[Sic10] Sick AG. *Sick Laser Sensor S300 Standard*, 2010.

[Sti86] S.M. Stigler. *The History of Statistics: The Measurement of Uncertainty Before 1900*. Belknap Series. Belknap Press of Harvard University Press, 1986.

[Sup08] Michael Suppa. *Autonomous Robot Work Cell Exploration Using Multisensory Eye-in-hand Systems*. PhD thesis, Universität Hannover, 2008.

[SW48] Claude Elwood Shannon and Warren Weaver. *A mathematical theory of communication*. American Telephone and Telegraph Company, 1948.

[TBF05] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, MA, 2005.

[TKS14] Ulrike Thomas, Simon Kriegel, and Michael Suppa. Fusing color and geometry information for understanding cluttered scenes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Robots in Clutter Workshop*, Chicago, Illinois, USA, September 14–18, 2014. IEEE.

[TPB06] R. Triebel, P. Pfaff, and W. Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, Beijing, China, October 9–15 2006. IEEE/RSJ, IEEE.

[Wal91] Peter Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman & Hall/CRC, 1991.

[WHB+10] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *Proceedings of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, AK, USA, May 2010. Software available at http://octomap.sf.net/.

[Woh15] Thomas Wohlfahrt. Exploration for autonomous 3d voxel mapping of static indoor environments with depth cameras and 2d odometry. Master's thesis, TUM, 2015.

[Yam97] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation, 1997, CIRA'97, Proceedings*, pages 146–151. IEEE, 1997.

Statutory Declaration

I, Alexandra von Lösecke, herewith declare that I have completed the present thesis independently making use only of the specified literature and aids. Sentences or parts of sentences quoted literally are marked as quotations; identification of other references with regard to the statement and scope of the work is quoted. I further declare that this work has not been submitted for credit elsewhere.

(place, date)	(signature)